



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2024년11월06일  
(11) 등록번호 10-2726547  
(24) 등록일자 2024년10월31일

(51) 국제특허분류(Int. Cl.)  
H04L 9/08 (2006.01) G06F 21/53 (2013.01)  
G06F 21/60 (2013.01) G06F 21/74 (2013.01)  
H04L 9/32 (2006.01)  
(52) CPC특허분류  
H04L 9/0822 (2013.01)  
G06F 21/53 (2013.01)  
(21) 출원번호 10-2021-7039137  
(22) 출원일자(국제) 2020년06월10일  
심사청구일자 2021년11월30일  
(85) 번역문제출일자 2021년11월29일  
(65) 공개번호 10-2022-0002616  
(43) 공개일자 2022년01월06일  
(86) 국제출원번호 PCT/EP2020/066133  
(87) 국제공개번호 WO 2020/260026  
국제공개일자 2020년12월30일  
(30) 우선권주장  
16/449,904 2019년06월24일 미국(US)  
(56) 선행기술조사문헌  
KR100904549 B1\*  
US20170251023 A1\*  
US20190058696 A1\*  
US20180109378 A1  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
인터내셔널 비지네스 머신즈 코포레이션  
미국 10504 뉴욕주 아몬크 뉴오차드 로드  
(72) 발명자  
로드리게즈, 에두아르도  
미국 텍사스 78758-3400, 오스틴, 버넷 로드 11501, 아이비엠 코포레이션  
카르나티, 프라딕  
미국 노스캐롤라이나 27703-9141, 더럼, 사우스 마이애미 불러버드 4205, 아이비엠 코포레이션  
보지레디, 카루나카르  
미국 텍사스 78758, 오스틴, 버넷 로드 11501, 인터랙티브 프로퍼티 로, 아이비엠 코포레이션  
(74) 대리인  
허정훈

전체 청구항 수 : 총 22 항

심사관 : 양종필

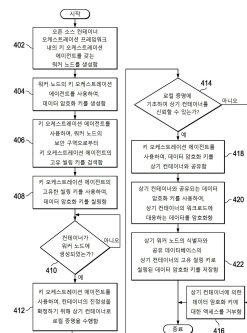
(54) 발명의 명칭 멀티-노드 클러스터에서 신뢰할 수 있는 컨테이너들 간의 암호화 키 오케스트레이션

(57) 요약

멀티-노드 클러스터에서 신뢰할 수 있는 컨테이너들 간의 암호화 키 오케스트레이션(CRYPTOGRAPHIC KEY ORCHESTRATION BETWEEN TRUSTED CONTAINERS IN A MULTI-NODE CLUSTER)

관리되는 컨테이너 환경에서 공유된 민감한 정보에 액세스하는 것이 제공된다. 복수의 워커 노드들의 각 워커 노드 (뒷면에 계속)

대표도 - 도4



드는 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역에 위치한 공유된 민감한 애플리케이션 데이터에 액세스할 수 있다. 상기 복수의 워커 노드들의 각 워커 노드는 각 워커 노드의 상기 보안 구역에서 고유한 셸링 키를 사용하여 각 워커 노드의 상기 데이터 암호화 키를 보호한다.

(52) CPC특허분류

*G06F 21/602* (2013.01)

*G06F 21/74* (2013.01)

*H04L 9/0894* (2013.01)

*H04L 9/3234* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

삭제

#### 청구항 2

관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법에 있어서, 상기 방법은:

복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared sensitive application data)를, 액세스하는 단계;

상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계;

클러스터(a cluster)에 제1 워커 노드를 설정하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를, 생성하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를 보호하기 위해 상기 제1 워커 노드 상의 상기 보안 구역에 고유한 제1 씰링 키를, 생성하는 단계;

상기 클러스터에서 제2 워커 노드의 생성에 응답하여, 상기 제1 워커 노드에 의해서, 상기 클러스터의 제2 워커 노드로 원격 증명(remote attestation)을, 수행하는 단계; 및

상기 원격 증명의 성공적 수행에 따라, 상기 제1 워커 노드에 의해서, 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을, 제공하는 단계를 포함하는

방법.

#### 청구항 3

제2항에 있어서, 상기 방법은:

상기 제2 워커 노드에 의해서, 상기 제2 워커 노드 상의 상기 보안 구역에 고유한 제2 씰링 키를, 생성하고, 상기 제2 씰링 키를 사용하여 데이터 암호화 키를 암호화함에 의해 상기 데이터 암호화 키를 보호하는 단계를 더 포함하는

방법.

#### 청구항 4

삭제

#### 청구항 5

관리되는 컨테이너 환경에서 공유된 민감한 정보에 액세스하기 위한 시스템에 있어서, 상기 시스템은:

복수의 워커 노드들을 포함하고;

상기 복수의 워커 노드들의 각 워커 노드는 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역에 위치한 공유된 민감한 애플리케이션 데이터를 액세스하며;

상기 복수의 워커 노드들의 각 워커 노드는 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키를 사용하여 각 워커 노드의 상기 데이터 암호화 키를 보호하며;

제1 워커 노드가 하나의 클러스터에 설정되고;

상기 제1 워커 노드는 상기 데이터 암호화 키를 생성하며;

상기 제1 워커 노드는 상기 데이터 암호화 키를 보호하기 위해 상기 제1 워커 노드 상에 상기 보안 구역에 대해서 고유한 제1 셸링 키를 생성하고;

상기 제1 워커 노드는 상기 클러스터의 제2 워커 노드의 생성에 응답하여 상기 클러스터의 제2 워커 노드로 원격 증명을 수행하며; 그리고

상기 제1 워커 노드는 상기 원격 증명의 성공적 수행에 따라 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을 제공하는

시스템.

## 청구항 6

제5항에 있어서:

상기 제2 워커 노드는 상기 제2 워커 노드 상의 보안 구역에 대해서 고유한 제2 셸링 키를 생성하고 상기 제2 셸링 키를 사용하여 상기 데이터 암호화 키를 암호화함에 의해서 상기 데이터 암호화 키를 보호하는

시스템.

## 청구항 7

삭제

## 청구항 8

구현된 프로그램 명령들을 갖는 컴퓨터 판독 가능 스토리지 매체에 있어서, 상기 프로그램 명령들은 한 세트의 프로세서들에 의해서 실행 가능하며, 상기 세트의 프로세서들이 관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법을 수행하게 하고, 상기 방법은:

복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터를, 액세스하는 단계;

상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 셸링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계;

하나의 클러스터(a cluster)에 제1 워커 노드를 설정하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를, 생성하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를 보호하기 위해 상기 제1 워커 노드 상의 상기 보안 구역에 대해서 고유한 제1 셸링 키를, 생성하는 단계;

상기 클러스터의 제2 워커 노드의 생성에 응답하여, 상기 제1 워커 노드에 의해서, 상기 클러스터의 제2 워커 노드로 원격 증명(remote attestation)을, 수행하는 단계; 및

상기 원격 증명의 성공적 수행에 따라, 상기 제1 워커 노드에 의해서, 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을, 제공하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

## 청구항 9

제8항에 있어서, 상기 방법은:

상기 제2 워커 노드에 의해서, 상기 제2 워커 노드 상의 보안 구역에 대해서 고유한 제2 셸링 키를 생성하고, 상기 제2 셸링 키를 사용하여 데이터 암호화 키를 암호화함에 의해 상기 데이터 암호화 키를 보호하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

## 청구항 10

암호화 키 오케스트레이션(cryptographic key orchestration)을 위한 컴퓨터-구현 방법에 있어서, 상기 컴퓨터-구현 방법은:

오픈 소스 컨테이너 오케스트레이션 프레임워크(an open source container orchestration framework)의 클러스터에 제1 키 오케스트레이션 에이전트(a first key orchestration agent)를 갖는 제1 워커 노드를, 한 세트의 프로세서들에 의해서, 생성하는 단계;

상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 생성하는 단계;

상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 쉘링 키를, 상기 프로세서들 세트에 의해서, 검색하는 단계;

상기 제1 키 오케스트레이션 에이전트의 상기 쉘링 키를 사용하여, 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 쉘링하는 단계를 포함하는

컴퓨터-구현 방법.

## 청구항 11

제10항에 있어서, 상기 방법은:

컨테이너가 상기 제1 워커 노드에 생성되었는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

컨테이너가 상기 제1 워커 노드에서 생성되었다고 결정하는 것에 응답하여, 상기 컨테이너의 진정성(authenticity)을 확립하기 위해(establish), 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너로 로컬 증명을, 상기 프로세서들 세트에 의해서, 수행하는 단계를 더 포함하는

컴퓨터-구현 방법.

## 청구항 12

제11항에 있어서, 상기 방법은:

상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 있는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 없다고 결정하는 것에 응답하여, 상기 컨테이너에 의한 상기 데이터 암호화 키에 대한 액세스를, 상기 프로세서들 세트에 의해서, 거부하는 단계를 더 포함하는

컴퓨터-구현 방법.

## 청구항 13

제12항에 있어서, 상기 방법은:

상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 있다고 결정하는 것에 응답하여, 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너와 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 공유하는 단계;

상기 컨테이너와 공유된 상기 데이터 암호화 키를 사용하여, 상기 컨테이너의 워크로드에 대응하는 데이터를, 상기 프로세서들 세트에 의해서, 암호화하는 단계; 및

상기 제1 워커 노드의 식별자와 공유된 데이터베이스의 상기 컨테이너의 고유한 쉘링 키로 쉘링된 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 저장하는 단계를 더 포함하는

컴퓨터-구현 방법.

## 청구항 14

제10항에 있어서, 상기 방법은:

상기 클러스터 내의 제2 키 오케스트레이션 에이전트를 갖는 제2 워커 노드를, 상기 프로세서들 세트에 의해서, 생성하는 단계; 및

상기 제2 키 오케스트레이션 에이전트의 진정성을 확인하기 위해, 상기 제2 워커 노드의 상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트로 원격 증명을, 상기 프로세서들 세트에 의해서, 수행하는 단계를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 15

제14항에 있어서, 상기 방법은:

상기 원격 증명에 기초하여 상기 제2 키 오케스트레이션 에이전트가 신뢰할 수 있는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

상기 원격 증명에 기초하여 상기 제2 키 오케스트레이션 에이전트가 신뢰할 수 없다고 결정하는 것에 응답하여, 상기 제2 키 오케스트레이션 에이전트에 의한 상기 제1 키 오케스트레이션 에이전트의 상기 데이터 암호화 키에 대한 액세스를, 상기 프로세서들 세트에 의해서, 거부하는 단계를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 16

제15항에 있어서, 상기 방법은:

상기 원격 증명에 기초하여 상기 제2 키 오케스트레이션 에이전트가 신뢰할 수 있다고 결정하는 것에 응답하여, 상기 제1 워커 노드의 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제2 키 오케스트레이션 에이전트와 상기 제1 키 오케스트레이션 에이전트의 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 공유하는 단계;

상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제2 워커 노드의 보안 구역으로부터 상기 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를, 상기 프로세서들 세트에 의해서, 검색하는 단계; 및

상기 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를 사용하여, 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 쉘링하는 단계를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 17

제10항에 있어서, 컨테이너들이 상기 클러스터 내에서 이동하였을 때에도, 민감한 애플리케이션 데이터는 신뢰할 수 있는 컨테이너들 사이에서만 공유되는

컴퓨터-구현 방법.

#### 청구항 18

제17항에 있어서, 상기 민감한 애플리케이션 데이터는 하드 디스크, 및 파일 스토리지에서 보호되는

컴퓨터-구현 방법.

#### 청구항 19

구현된 프로그램 명령들을 갖는 컴퓨터 판독 가능 스토리지 매체에 있어서, 상기 프로그램 명령들은 한 세트의 프로세서들에 의해서 실행 가능하며, 상기 세트의 프로세서들이 암호화 키 오케스트레이션을 위한 방법을 수행하게 하고, 상기 방법은:

오픈 소스 컨테이너 오케스트레이션 프레임워크의 클러스터에 제1 키 오케스트레이션 에이전트를 갖는 제1 워커 노드를, 상기 프로세서들 세트에 의해서, 생성하는 단계;

상기 제1 워커 노드의 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 생성하는 단계;

상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 쉘링 키를, 상기 프로세서들 세트에 의해서, 검색하는 단계; 및

상기 제1 키 오케스트레이션 에이전트의 쉘링 키를 사용하여, 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 쉘링하는 단계를 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 20

제19항에 있어서, 상기 방법은:

하나의 컨테이너가 상기 제1 워커 노드에서 생성되었는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

하나의 컨테이너가 상기 제1 워커 노드에서 생성되었다고 결정하는 것에 응답하여, 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너의 진정성을 확인하기 위해 상기 컨테이너로 로컬 증명(local attestation)을, 상기 프로세서들 세트에 의해서, 수행하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 21

제20항에 있어서, 상기 방법은:

상기 컨테이너를 상기 로컬 증명에 기초하여 신뢰할 수 있는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

상기 컨테이너를 상기 로컬 증명에 기초하여 신뢰할 수 없다고 결정하는 것에 응답하여, 상기 컨테이너에 의한 상기 데이터 암호화 키에 대한 액세스를, 상기 프로세서들 세트에 의해서, 거부하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 22

제21항에 있어서, 상기 방법은:

상기 컨테이너를 상기 로컬 증명에 기초하여 신뢰할 수 있다고 결정하는 것에 응답하여, 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너와 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 공유하는 단계;

상기 컨테이너와 함께 공유된 상기 데이터 암호화 키를 사용하여, 상기 컨테이너의 워크로드에 대응하는 데이터를, 상기 프로세서들 세트에 의해서, 암호화하는 단계; 및

상기 제1 워커 노드의 식별자와 공유된 데이터베이스의 상기 컨테이너의 고유한 쉘링 키로 쉘링된 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 저장하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 23

제19항에 있어서, 상기 방법은:

상기 클러스터 내에 제2 키 오케스트레이션 에이전트를 갖는 제2 워커 노드를, 상기 프로세서들 세트에 의해서, 생성하는 단계; 및

상기 제2 키 오케스트레이션 에이전트의 진정성을 확인하기 위해, 상기 제2 워커 노드의 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 제1 키 오케스트레이션 에이전트로 원격 증명을, 상기 프로세서들 세트에 의해서, 수행하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 24

제23항에 있어서, 상기 방법은:

상기 제2 키 오के스트레이션 에이전트를 상기 원격 증명에 기초하여 신뢰할 수 있는지를, 상기 프로세서들 세트에 의해서, 결정하는 단계; 및

상기 제2 키 오케스트레이션 에이전트를 상기 원격 증명에 기초하여 신뢰할 수 없다고 결정하는 것에 응답하여, 상기 제2 키 오케스트레이션 에이전트에 의한 상기 제1 키 오케스트레이션 에이전트의 데이터 암호화 키에 대한 액세스를, 상기 프로세서들 세트에 의해서, 거부하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 25

제24항에 있어서, 상기 방법은:

상기 제2 키 오케스트레이션 에이전트를 상기 원격 증명에 기초하여 신뢰할 수 있다고 결정하는 것에 응답하여, 상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제2 키 오케스트레이션 에이전트와 상기 제1 키 오케스트레이션 에이전트의 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 공유하는 단계;

상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제2 워커 노드의 보안 구역으로부터 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를, 상기 프로세서들 세트에 의해서, 검색하는 단계; 및

상기 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를 사용하여, 상기 데이터 암호화 키를, 상기 프로세서들 세트에 의해서, 쉘링하는 단계를 더 포함하는

컴퓨터 판독 가능 스토리지 매체.

#### 청구항 26

삭제

#### 청구항 27

삭제

#### 청구항 28

삭제

#### 청구항 29

삭제

#### 청구항 30

삭제

#### 청구항 31

삭제

#### 청구항 32

삭제

#### 청구항 33

삭제

#### 청구항 34

삭제

## 청구항 35

삭제

### 발명의 설명

#### 기술 분야

[0001] 본 설명서는 일반적으로 오픈 소스 컨테이너 오케스트레이션 프레임워크들(open source container orchestration frameworks)와 관련이 있으며, 더 구체적으로는 오픈 소스 컨테이너 결합 프레임워크의 멀티-노드 클러스터 내에서 신뢰할 수 있는 노드들과 컨테이너들(trusted nodes and containers) 사이에서 암호와 키 오케스트레이션(cryptographic key orchestration)을 제공하는 것에 관한 것이다.

#### 배경 기술

[0002] 예를 들어, 쿠버네티스(Kubernetes)와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크는 상호 연결된 물리적 및/또는 가상 컴퓨팅 노드들(즉, 머신들)의 클러스터이며, 이는 여기서 컨테이너화 된 애플리케이션들(containerized applications)의 자동 배치(automatic deployment), 확장(scaling) 및 관리를 제공한다. 오픈 소스 컨테이너 오케스트레이션 프레임워크는 애플리케이션 워크로드들(application workloads)을 대신하여 컴퓨팅, 네트워킹 및 스토리지 인프라스트럭처를 오케스트레이트한다(orchestrate). 이것은 서비스형 인프라스트럭처(Infrastructure as a Service: IaaS)의 유연성과 함께 서비스형 플랫폼(Platform as a Service: PaaS)의 단순성의 많은 부분을 제공하며, 인프라스트럭처 공급자들 간의 이식성(portability across infrastructure providers)을 가능하게 한다.

[0003] 노드들의 클러스터는, 마스터 노드 및 워커 노드(a master node and a worker node), 두 가지 유형의 노드들을 포함한다. 마스터 노드는 상기 클러스터의 워커 노드들을 관리하는 것을 담당한다. 마스터 노드는, 예를 들어, 애플리케이션 프로그래밍 인터페이스, 사용자 인터페이스, 또는 커맨드 라인 인터페이스로부터 수신되는, 상기 클러스터에서 실행될 모든 연산들의 엔트리 포인트(the entry point for all operations)이다. 워커 노드는 하나 또는 그 이상의 워크로드들을 실행하는 것과 상기 클러스터의 다른 워커 노드들에서 실행되는 서로 다른 워크로드들과의 네트워킹하는 것을 담당한다. 오픈 소스 컨테이너 오케스트레이션 프레임워크의 특징들은, 예를 들어, 빠른 애플리케이션 배치, 자동-복구(auto-recovery) 및 자가-치유(self-healing) 및 원활한 애플리케이션 갱신(seamless application update)을 포함할 수 있다.

### 발명의 내용

#### 해결하려는 과제

[0004] 워커 노드의 기초 스케줄링 유닛(The basic scheduling unit)은, 예를 들어, 포드(a pod)와 같은, 컨테이너이다. 컨테이너들은 워커 노드 상에 함께-위치되고(co-located) 자원들을 공유할 수 있다. 오픈 소스 컨테이너 오케스트레이션 프레임워크의 각 컨테이너는 워커 노드들의 클러스터 내의 고유 컨테이너 인터넷 프로토콜 주소가 할당되는데, 이는 애플리케이션들이 충돌의 위험 없이 포트들을 사용할 수 있게 한다. 현재, 컨테이너들은 호스트 머신에서 서로 참조할 수는 있지만, 하나의 워커 노드에 위치한 컨테이너는 다른 워커 노드 내의 다른 컨테이너에 직접 주소를 지정할 방법이 없으므로, 컨테이너의 고유 인터넷 프로토콜 주소를 사용할 수밖에 없다. 컨테이너는, 예를 들어, 로컬 디스크 디렉토리 또는 네트워크 디스크와 같은, 볼륨(a volume)을 정의하고, 상기 볼륨을 워커 노드 내의 다른 컨테이너들에 노출시킬 수 있다. 서비스는 다수-계층(multi-tier) 애플리케이션의 한 단계와 같이, 함께 작업하는 한 세트의 컨테이너들이다.

[0005] 그러나, 다수의 워커 노드들을 포함하는 클러스터에서, 상기 클러스터 상에서 실행하는 애플리케이션들 사이에서 민감한(sensitive) 데이터를 액세스하는 것은 현재 가능하지 않다. 민감한 애플리케이션 데이터가 워커 노드의 보안 구역에 의해 쉴링되었을 때, 특히 그렇다. 보안 구역은 애플리케이션 기능의 중요한 부분들이 보호되는 격리된 신뢰할 수 있는 메모리의 영역이며, 이 것은 민감한 애플리케이션 데이터를 기밀로 유지하고 수정하지 않도록 돕는다. 바꾸어 말하면, 보안 구역은 메인 메모리의 암호화된 부분이다. 보안 구역은 프로세서-기반 보안 메커니즘들에 의해 강화된다. 주어진 머신의 하드웨어(예: 워커 노드의 프로세서 칩)으로부터 유래된 쉴링된 암호화 키에 보안을 태그하는 것은 보안성은 높지만, 이것은 현재, 쿠버네티스와 같은, 동적인 분산된 오픈 소스 컨테이너 오케스트레이션 프레임워크에서, 민감한 애플리케이션 데이터를 공유하는 방법이며,

비록 해결이 불가능한 것은 아니지만, 많은 문제들이 존재한다.

## 과제의 해결 수단

- [0006] 본 발명의 일 예시적 실시 예에 따라, 관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법이 제공된다. 복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드는 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared sensitive application data)에 액세스한다. 상기 복수의 워커 노드들의 각 워커 노드는 각 워커 노드 상의 상기 보안 구역에 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를 보호한다. 또한, 상기 방법은 클러스터(a cluster)에 제1 워커 노드를 설정한다. 상기 제1 워커 노드는 상기 데이터 암호화 키와 상기 데이터 암호화 키를 보호하기 위한 상기 제1 워커 노드 상의 상기 보안 구역에 고유한 제1 씰링 키를 생성한다. 상기 제1 워커 노드는 상기 클러스터에서 상기 제2 워커 노드의 생성에 응답하여 상기 클러스터의 제2 워커 노드로 원격 증명(remote attestation)을 수행한다. 상기 제1 워커 노드는 상기 원격 증명의 성공적 수행에 따라 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을 제공한다. 또한, 상기 제2 워커 노드는 상기 제2 워커 노드 상의 상기 보안 구역에 고유한 제2 씰링 키를 생성하고, 상기 제2 씰링 키를 사용하여 데이터 암호화 키를 암호화함에 의해 상기 데이터 암호화 키를 보호한다. 본 발명의 다른 예시적 실시 예에 따라, 관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 시스템 및 컴퓨터 프로그램 제품이 제공된다.
- [0007] 본 발명의 또 다른 예시적 실시 예에 따라, 암호화 키 오케스트레이션(cryptographic key orchestration)을 위한 컴퓨터-구현 방법이 제공된다. 한 세트의 프로세서들은 오픈 소스 컨테이너 오케스트레이션 프레임워크(an open source container orchestration framework)의 클러스터에 제1 키 오케스트레이션 에이전트(a first key orchestration agent)를 갖는 제1 워커 노드를 생성한다. 상기 세트의 프로세서들은, 상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를 생성한다. 상기 세트의 프로세서들은, 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 씰링 키를 검색한다(retrieve). 상기 세트의 프로세서들은, 상기 제1 키 오케스트레이션 에이전트의 상기 씰링 키를 사용하여, 상기 데이터 암호화 키를 씰링한다(seal). 또한, 상기 세트의 프로세서들은 컨테이너가 상기 제1 워커 노드에 생성되었는지를 결정한다. 컨테이너가 상기 제1 워커 노드에서 생성되었다고 결정하는 것에 응답하여, 상기 세트의 프로세서들은 상기 컨테이너의 진정성(authenticity)을 확립하기 위해(establish), 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너로 로컬 증명을 수행한다. 그 다음, 상기 세트의 프로세서들은 상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 있는지를 결정한다. 상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 없다고 결정하는 것에 응답하여, 상기 세트의 프로세서들은 상기 컨테이너에 의한 상기 데이터 암호화 키에 대한 액세스를, 거부한다(deny). 상기 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 있다고 결정하는 것에 응답하여, 상기 세트의 프로세서들은 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너와 상기 데이터 암호화 키를, 공유하고(share), 상기 컨테이너와 공유된 상기 데이터 암호화 키를 사용하여, 상기 컨테이너의 워크로드에 대응하는 데이터를, 암호화하며(encrypt), 그리고 상기 제1 워커 노드의 식별자와 공유된 데이터베이스의 상기 컨테이너의 고유한 씰링 키로 씰링된 상기 데이터 암호화 키를 저장한다(store). 본 발명의 또 다른 실시 예에 따라, 암호화 키 오케스트레이션을 위한 컴퓨터 프로그램 제품이 제공된다.
- [0008] 그 결과, 본 발명의 예시적 실시 예들은 클러스터 내의 신뢰할 수 있는 컨테이너 사이에서 암호화 키 오케스트레이션을 제공할 수 있다. 또한, 상기 예시적 실시 예들은 상기 클러스터 내의 다른 워커 노드들 상의 분산 애플리케이션을 실행하는 다른 컨테이너들이 고유한 씰링 키로 씰링된 데이터 암호화 키를 사용하여 상기 다른 컨테이너들 사이에서 민감한 애플리케이션 데이터를 공유할 수 있게 한다.
- [0009] 일 실시 예에 따라, 관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법이 제공되고, 상기 방법은: 복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared sensitive application data)를, 액세스하는 단계; 및 상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계를 포함한다.
- [0010] 다른 실시 예에 따라, 관리되는 컨테이너 환경에서 공유된 민감한 정보에 액세스하기 위한 시스템이 제공되고, 상기 시스템은: 복수의 워커 노드들을 포함하고: 상기 복수의 워커 노드들의 각 워커 노드는 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared sensitive application data)에 액세스하는 단계; 및 상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계를 포함한다.

호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역에 위치한 공유된 민감한 애플리케이션 데이터를 액세스하며; 그리고 상기 복수의 워커 노드들의 각 워커 노드는 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 쉼 키를 사용하여 각 워커 노드의 상기 데이터 암호화 키를 보호한다.

[0011] 또 다른 실시 예에 따라, 관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 컴퓨터 프로그램 제품이 제공되고, 상기 컴퓨터 프로그램 제품은 구현된 프로그램 명령들을 갖는 컴퓨터 판독 가능 스토리지 매체를 포함하며, 상기 프로그램 명령들은 한 세트의 프로세서들에 의해서 실행 가능하고, 상기 세트의 프로세서들이 방법을 수행하게 하며, 상기 방법은: 복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터를, 액세스하는 단계; 및 상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 쉼 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계를 포함한다.

[0012] 또 다른 실시 예에 따라, 암호화 키 오케스트레이션(cryptographic key orchestration)을 위한 컴퓨터-구현 방법이 제공되고, 상기 컴퓨터-구현 방법은: 오픈 소스 컨테이너 오케스트레이션 프레임워크(an open source container orchestration framework)의 클러스터에 제1 키 오케스트레이션 에이전트(a first key orchestration agent)를 갖는 제1 워커 노드를, 생성하는 단계; 상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를, 생성하는 단계; 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 쉼 키를, 검색하는 단계; 상기 제1 키 오케스트레이션 에이전트의 상기 쉼 키를 사용하여, 상기 데이터 암호화 키를, 쉼하는 단계를 포함한다.

[0013] 또 다른 실시 예에 따라, 암호화 키 오케스트레이션을 위한 컴퓨터 프로그램 제품이 제공되고, 상기 컴퓨터 프로그램 제품은 구현된 프로그램 명령들을 갖는 컴퓨터 판독 가능 스토리지 매체를 포함하고, 상기 프로그램 명령들은 한 세트의 프로세서들에 의해서 실행 가능하며, 상기 세트의 프로세서들이 방법을 수행하게 하고, 상기 방법은: 오픈 소스 컨테이너 오케스트레이션 프레임워크의 클러스터에 제1 키 오케스트레이션 에이전트를 갖는 제1 워커 노드를, 생성하는 단계; 상기 제1 워커 노드의 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를, 생성하는 단계; 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 쉼 키를, 검색하는 단계; 및 상기 제1 키 오케스트레이션 에이전트의 쉼 키를 사용하여, 상기 데이터 암호화 키를, 쉼하는 단계를 포함한다.

[0014] 또 다른 실시 예에 따라, 암호화 키 오케스트레이션을 위한 시스템이 제공되고, 상기 시스템은 방법을 수행하도록 작동 가능하고, 상기 방법은: 오픈 소스 컨테이너 오케스트레이션 프레임워크의 클러스터에서 제1 키 오케스트레이션 에이전트를 갖는 제1 워커 노드를, 생성하는 단계; 상기 제1 워커 노드의 제1 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를, 생성하는 단계; 상기 제1 키 오케스트레이션 에이전트를 사용하여, 상기 제1 워커 노드의 보안 구역으로부터 상기 제1 키 오케스트레이션 에이전트의 쉼 키를, 검색하는 단계; 및 상기 제1 키 오케스트레이션 에이전트의 쉼 키를 사용하여, 상기 데이터 암호화 키를, 쉼하는 단계를 포함한다.

## 도면의 간단한 설명

[0015] 본 발명의 실시 예들은 다음과 같은 첨부 도면들과 참조하여, 예로서 만, 설명될 것이다:

도 1은 예시적 실시 예들이 구현될 수 있는 데이터 처리 시스템들의 네트워크의 개념도이다;

도 2는 예시적 실시 예들이 구현될 수 있는 데이터 처리 시스템의 다이어그램이다;

도 3은 예시적 실시 예에 따라 암호화 키 오케스트레이션 시스템의 일 예를 예시하는 다이어그램이다;

도 4는 예시적 실시 예에 따라 워커 노드와 워커 노드에 생성된 컨테이너 사이에서 암호화 키 오케스트레이션을 위한 프로세스를 예시하는 플로차트다;

도 5a-5b는 예시적 실시 예에 따라 클러스터 내에서 기존 워커 노드와 새로 발생된 워커 노드 사이에서 암호화 키 오케스트레이션을 위한 프로세스를 예시하는 플로차트다;

도 6은 예시적 실시 예에 따라 워커 노드들의 데이터 암호화 키들을 보호하기 위한 프로세스를 예시하는 플로차트다; 및

도 7은 예시적 실시 예에 따라 클러스터의 워커 노드들 사이에서 데이터 암호화 키를 공유하기 위한 프로세스를 예시하는 플로차트다.

### 발명을 실시하기 위한 구체적인 내용

- [0016] 본 발명의 실시 예들은 시스템, 방법, 및/또는 통합의 모든 가능한 기술적 세부 수준에서 컴퓨터 프로그램 제품이 될 수 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 스토리지 매체를 포함할 수 있으며, 이 매체 상에 프로세서가 본 발명의 실시 예들을 수행하도록 하는 컴퓨터 판독 가능 프로그램 명령들을 갖는다.
- [0017] 상기 컴퓨터 판독 가능 스토리지 매체는 명령 실행 장치에 의해 사용될 명령들을 유지 및 저장할 수 있는 유형의(tangible) 디바이스일 수 있다. 상기 컴퓨터 판독 가능 스토리지 매체는, 예를 들면, 전자 스토리지 디바이스, 자기 스토리지 디바이스, 광 스토리지 디바이스, 전자기 스토리지 디바이스, 반도체 스토리지 디바이스, 또는 전술한 것들의 모든 적절한 조합일 수 있으며, 그러나 이에 한정되지는 않는다. 컴퓨터 판독 가능 스토리지 매체의 더 구체적인 예들의 비포괄적인 목록에는 다음이 포함될 수 있다: 휴대용 컴퓨터 디스켓, 하드 디스크, 랜덤 액세스 메모리(RAM), 판독-전용 메모리(ROM), 소거 및 프로그램가능 판독-전용 메모리(EPROM 또는 플래시 메모리), 정적 랜덤 액세스 메모리(SRAM), 휴대용 콤팩트 디스크 판독-전용 메모리(CD-ROM), 디지털 다용도 디스크(DVD), 메모리 스틱, 플로피 디스크, 천공-카드들 또는 명령들이 기록된 홈에 있는 융기된 구조들 같이 기계적으로 인코드 된 장치, 및 전술한 것들의 모든 적절한 조합. 본 명세서에서 사용될 때, 컴퓨터 판독 가능 스토리지 매체는 무선 전파들이나 다른 자유롭게 전파되는 전자기파들, 도파관이나 기타 전송 매체(예를 들어, 광섬유 케이블을 통해 전달되는 광 펄스들)를 통해 전파되는 전자기파들, 또는 선(wire)을 통해 전송되는 전기 신호들 같이 그 자체로 일시적인(transitory) 신호들로 해석되지는 않는다.
- [0018] 본 명세서에 기술되는 컴퓨터 판독 가능 명령들은, 예를 들어, 인터넷, 근거리 통신망, 광역 통신망 및/또는 무선 네트워크 등의 통신망(네트워크)을 통해 컴퓨터 판독 가능 스토리지 매체로부터 각각 컴퓨팅/처리 디바이스들로 또는 외부 스토리지 디바이스로부터 외부 컴퓨터로 다운로드 될 수 있다. 상기 통신망은 구리 전송 케이블들, 광 전송 섬유들, 무선 전송, 라우터들, 방화벽들, 스위치들, 게이트웨이 컴퓨터들 및/또는 엣지 서버들을 포함할 수 있다. 각 컴퓨팅/처리 장치 내 네트워크 어댑터 카드 또는 네트워크 인터페이스는 상기 통신망으로부터 컴퓨터 판독 가능 프로그램 명령들을 수신하고 그 컴퓨터 판독 가능 프로그램 명령들을 각각의 컴퓨팅/처리 디바이스 내의 컴퓨터 판독 가능 스토리지 매체에 저장하기 위해 포워드한다.
- [0019] 본 발명의 연산들을 실행하기 위한 컴퓨터 판독 가능 프로그램 명령들은 Smalltalk, C++ 또는 그와 유사 언어 등의 객체 지향 프로그래밍 언어와 "C" 프로그래밍 언어 또는 그와 유사한 프로그래밍 언어 등의 종래의 절차적 프로그래밍 언어들을 포함하여, 하나 또는 그 이상의 프로그래밍 언어들을 조합하여 작성된(written) 어셈블리 명령들, 명령-세트-아키텍처(ISA) 명령들, 머신 명령들, 머신 종속 명령들, 마이크로코드, 펌웨어 명령들, 상태-셋팅 데이터, 집적회로를 위한 구성 데이터, 또는 소스 코드나 목적 코드일 수 있다. 상기 컴퓨터 판독 가능 프로그램 명령들은 전적으로 사용자의 컴퓨터상에서, 부분적으로 사용자의 컴퓨터상에서, 독립형(stand-alone) 소프트웨어 패키지로서, 부분적으로 사용자의 컴퓨터상에서 그리고 부분적으로 원격 컴퓨터상에서 또는 전적으로 원격 컴퓨터나 서버상에서 실행될 수 있다. 위에서 마지막의 경우에, 원격 컴퓨터는 근거리 통신망(LAN) 또는 광역 통신망(WAN)을 포함한 모든 종류의 네트워크를 통해서 사용자의 컴퓨터에 접속될 수 있고, 또는 이 접속은 (예를 들어, 인터넷 서비스 제공자를 이용한 인터넷을 통해서) 외부 컴퓨터에 이루어질 수도 있다. 일부 실시 예들에서, 예를 들어 프로그램 가능 로직 회로, 필드-프로그램 가능 게이트 어레이들(FPGA), 또는 프로그램 가능 로직 어레이들(PLA)을 포함한 전자 회로는 본 발명의 실시 예들을 수행하기 위해 전자 회로를 맞춤화하도록 상기 컴퓨터 판독 가능 프로그램 명령들의 상태 정보를 활용하여 상기 컴퓨터 판독 가능 프로그램 명령들을 실행할 수 있다.
- [0020] 본 명세서에서는 본 발명의 실시 예들에 따른 방법들, 장치들(시스템들), 및 컴퓨터 프로그램 제품들의 플로 차트 예시도들 및/또는 블록도들을 참조하여 본 발명의 실시 예들을 기술한다. 플로 차트 예시도들 및/또는 블록도들의 각 블록과 플로 차트 예시도들 및/또는 블록도들 내 블록들의 조합들은 컴퓨터 판독 가능 프로그램 명령들에 의해 구현될 수 있다는 것을 이해할 수 있을 것이다.
- [0021] 이들 컴퓨터 판독 가능 프로그램 명령들은 범용 컴퓨터, 특수목적용 컴퓨터, 또는 기타 프로그램가능 데이터 처리 장치의 프로세서에 제공되어 머신(machine)을 생성하고, 그렇게 하여 그 명령들이 상기 컴퓨터 또는 기타 프로그램가능 데이터 처리 장치의 프로세서를 통해서 실행되어, 상기 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/동작들을 구현하기 위한 수단을 생성할 수 있다. 이들 컴퓨터 판독 가능 프로그램

명령들은 또한 컴퓨터 판독 가능 스토리지 매체에 저장될 수 있으며, 컴퓨터, 프로그램가능 데이터 처리 장치 및/또는 기타 디바이스들에 지시하여 명령들이 저장된 상기 컴퓨터 판독 가능 스토리지 매체가 상기 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능/동작의 특징들을 구현하는 명령들을 포함하는 제조품(an article of manufacture)을 포함하도록 특정한 방식으로 기능하게 할 수 있다.

[0022] 상기 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터, 기타 프로그램가능 데이터 처리 장치, 또는 다른 디바이스에 로드 되어, 상기 컴퓨터, 기타 프로그램가능 장치 또는 다른 디바이스에서 일련의 동작 단계들이 수행되게 하여 컴퓨터 구현 프로세스를 생성하며, 그렇게 하여 상기 컴퓨터, 기타 프로그램가능 장치, 또는 다른 디바이스 상에서 실행되는 명령들이 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/동작들을 구현할 수 있다.

[0023] 도면들 내 플로 차트 및 블록도들은 본 발명의 여러 실시 예들에 따른 시스템들, 방법들 및 컴퓨터 프로그램 제품들의 가능한 구현들의 아키텍처, 기능(functionality), 및 연산(operation)을 예시한다. 이와 관련하여, 상기 플로 차트 또는 블록도들 내 각 블록은 상기 명시된 논리적 기능(들)을 구현하기 위한 하나 또는 그 이상의 실행 가능한 명령들을 포함한 모듈, 세그먼트 또는 명령들의 일부분을 나타낼 수 있다. 일부 다른 구현들에서, 상기 블록에 언급되는 기능들은 도면들에 언급된 순서와 다르게 일어날 수도 있다. 예를 들면, 연속으로 도시된 두 개의 블록들은 실제로는 사실상 동시에 실행될 수도 있고, 또는 이 두 블록들은 때때로 관련된 기능에 따라서는 역순으로 실행될 수도 있다. 블록도들 및/또는 순서 예시도의 각 블록, 및 블록도들 및/또는 순서 예시도 내 블록들의 조합들은 특수목적용 하드웨어 및 컴퓨터 명령들의 명시된 기능들 또는 동작들, 또는 이들의 조합들을 수행하는 특수목적용 하드웨어-기반 시스템들에 의해 구현될 수 있다는 것에 또한 주목해야 한다.

[0024] 도면들 참조, 특히, 도 1-3을 참조하여, 데이터 처리 환경들의 다이어그램들이 제공되며, 여기서 예시적 실시 예들이 구현될 수 있다. 도 1-3은 오직 예들일 뿐이고 환경들과 관련하여 어떠한 제한도 주장하거나 암시하기 위한 것이 아니며, 서로 다른 실시 예들이 구현될 수 있음을 이해해야 한다. 도시된 환경들에 대한 많은 수정들이 이루어질 수 있다.

[0025] 도 1은 데이터 처리 시스템들의 네트워크의 개념도로 도시된 것이며, 여기서 예시적 실시 예들이 구현될 수 있다. 네트워크 데이터 처리 시스템(100)은 컴퓨터들, 데이터 처리 시스템들, 그리고 기타 디바이스들의 네트워크이며, 여기서 예시적 실시 예들이 구현될 수 있다. 네트워크 데이터 처리 시스템(100)은 네트워크(102)를 포함하며, 이는 네트워크 데이터 처리 시스템(100) 내에서 함께 연결된 컴퓨터들, 데이터 처리 시스템들 및 기타 디바이스들 사이의 통신 링크들을 제공하는 데 사용되는 매체이다. 네트워크(102)는, 예를 들어, 유선 통신 링크들, 무선 통신 링크들 및 광섬유 케이블들과 같은, 연결들을 포함할 수 있다.

[0026] 도시된 예에서, 서버(104)와 서버(106)는 스토리지(108)를 따라 네트워크(102)에 연결된다. 서버(104)와 서버(106)는, 예를 들어, 오픈 소스 컨테이너 오케스트레이션 프레임워크(112)의 클러스터(110) 내의 워커 노드들(worker nodes)일 수 있다. 오픈 소스 컨테이너 오케스트레이션 프레임워크(112)는, 예를 들어, 쿠버네티스 관리된 컨테이너 환경(Kubernetes managed container environment)일 수 있다. 서버(104)와 서버(106)는 물리적 서버들 또는 가상 서버들을 나타낼 수 있다. 또한, 서버(104)와 서버(106)는 네트워크(102)에 고속(high-speed) 연결들을 가질 수 있고, 하나 또는 그 이상의 분산된, 다수-계층(multi-tier) 애플리케이션들을 사용하여, 한 세트의 서비스들을 클라이언트 디바이스들에 제공할 수 있는데, 그러한 서비스들은, 예를 들어, 이벤트 모니터링 서비스들, 금융 서비스들, 은행 서비스들, 정부 서비스들, 교육 서비스들, 예약 서비스들, 데이터 서비스들 등을 포함한다. 또한, 서버(104)와 서버(106)는 각각 하나 또는 그 이상의 클라우드 환경들에서 컴퓨팅 노드들의 클러스터들을 대표할 수 있다. 이와 달리, 서버(104)와 서버(106)는 각각 하나 또는 그 이상의 데이터 센터들에 있는 서버들의 클러스터들을 대표할 수 있다.

[0027] 클라이언트(114), 클라이언트(116) 및 클라이언트(118)는 네트워크(102)에 또한 연결된다. 클라이언트들(114, 116 및 118)들은 서버(104)와 서버(106)의 클라이언트들이다. 이 예에서, 클라이언트들(114, 116 및 118)들은 네트워크(102)에 대한 유선 통신 링크들을 갖는 데스크톱 또는 개인용 컴퓨터들로 도시된다. 그러나, 클라이언트들(114, 116 및 118)들은 예들일 뿐이고, 예를 들어, 네트워크 컴퓨터들, 랩톱 컴퓨터들, 핸드헬드 컴퓨터들, 스마트폰들, 스마트 워치들, 스마트 텔레비전들, 스마트 어플라이언스들, 게임 디바이스들, 키오스크들 등과 같은, 다른 유형들의 데이터 처리 시스템들을 나타낼 수 있다. 클라이언트들(114, 116 및 118)의 이용자들은 클라이언트들(114, 116 및 118)을 이용하여 서버(104)와 서버(106)에 의해 제공되는 서비스들에 액세스하고 이용할 수 있다.

- [0028] 스토리지(108)는 구조화된 형식 또는 비구조화된 형식의 모든 유형의 데이터를 저장할 수 있는 네트워크 스토리지 디바이스다. 또한, 스토리지(108)는 복수의 네트워크 스토리지 디바이스들을 나타낼 수 있다. 또한, 스토리지(108)는 복수의 서버들을 위한 식별자들과 네트워크 주소들, 암호화 키들, 암호화된 데이터 등을 저장할 수 있다. 더욱이, 스토리지(108)는 인증 또는 자격 증명(credential) 데이터와 같은, 다른 유형들의 데이터를 저장할 수 있으며, 상기 인증 또는 자격 증명 데이터는 사용자 이름들, 패스워드들 및 시스템 관리자 및 사용자들과 관련된 생체 인식 데이터를 포함할 수 있다.
- [0029] 또한, 네트워크 데이터 처리 시스템(100)은 추가 서버들, 클라이언트들, 스토리지 디바이스들 및 도시되지 않은 다른 디바이스들을 포함할 수 있음을 유의해야 한다. 네트워크 정보처리 시스템(100)에 위치한 프로그램 코드는 컴퓨터 판독 가능 저장 매체에 저장되고 사용을 위해 컴퓨터 또는 다른 데이터 처리 디바이스로 다운로드 될 수 있다. 예를 들어, 프로그램 코드는 서버(104)의 컴퓨터 판독 가능 저장 매체에 저장되고 클라이언트(114) 상에서 사용하기 위해 네트워크(102)를 통해 클라이언트(114)로 다운로드 될 수 있다.
- [0030] 도시된 예에서, 네트워크 데이터 처리 시스템(100)은, 예를 들어, 인터넷, 인트라넷, 근거리 통신망(LAN), 광역 통신망(WAN), 통신 네트워크, 또는 이들의 모든 조합과 같은, 서로 다른 유형들의 통신 네트워크들로 구현될 수 있다. 도 1은 예시일 뿐이고, 다른 예시적 실시 예들에 대한 아키텍처를 한정하는 것이 아니다.
- [0031] 이제 도 2를 참조하며, 데이터 처리 시스템의 다이어그램이 예시적 일 실시 예에 따라 도시된다. 데이터 처리 시스템(200)은 도 1의 서버(104), 서버(106)와 같은, 컴퓨터의 예이며, 상기 컴퓨터에는 컴퓨터 판독 가능 프로그램 코드 또는 예시적 실시 예들의 프로세스를 구현하는 명령들이 위치할 수 있다. 데이터 처리 시스템(200)은, 예를 들어, 쿠버네티스와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크의 워커 노드들의 한 클러스터 내의 하나의 워커 노드일 수 있다. 데이터 처리 시스템(200)은 한 세트의 자원들을 포함하고 컨트롤하며, 상기 한 세트의 자원들은 데이터 처리 시스템(200)에 의해서 사용될 수 있다. 이 예시적 예에서, 데이터 처리 시스템(200)은 통신 패브릭(202)을 포함하며, 상기 통신 패브릭(202)은 프로세서 유닛(204), 메모리(206), 영구 스토리지(208), 통신 유닛(210), 입/출력(I/O) 유닛(212) 및 디스플레이(214) 사이의 통신들을 제공한다. 그러나, 데이터 처리 시스템(200)은 예일뿐이고 예시된 것보다 더 많거나 더 적은 컴포넌트들을 포함할 수 있다는 점에 유의해야 한다.
- [0032] 프로세서 유닛(204)은 소프트웨어 애플리케이션들 및 프로그램들의 명령들을 실행하게 하며, 상기 소프트웨어 애플리케이션들 및 프로그램들의 명령들은 메모리(206)에 로드 될 수 있다. 프로세서 유닛(204)은, 특정 구현에 따라, 한 세트의 하나 또는 그 이상의 하드웨어 프로세서 디바이스들이 될 수 있고 또는 멀티-코어(multi-core) 프로세서일 수 있다.
- [0033] 메모리(206) 및 영구 스토리지(208)는 스토리지 디바이스의 예들이다. 컴퓨터 판독 가능 스토리지 디바이스는 정보를 저장할 수 있는 모든 종류의 하드웨어이며, 상기 정보는, 예를 들어, 데이터, 기능적 형태의 컴퓨터 판독 가능 프로그램 코드 및/또는 일시적 기반 및/또는 영구적 기반 둘 중 하나의 기타 적절한 정보를 포함한다. 또한, 컴퓨터 판독 가능 스토리지 디바이스는 전파 매체(a propagation medium)를 제외한다. 이들 예들에서, 메모리(206)는, 예를 들어, 랜덤-액세스 메모리(RAM) 또는 모든 기타 적절한 휘발성 또는 비휘발성(non-volatile) 스토리지 디바이스일 수 있다. 영구 스토리지(208)는, 특정 구현 예에 따라(depending), 다양한 형태들을 취할 수 있다. 예를 들어, 영구 스토리지(208)는 하나 또는 그 이상의 디바이스들을 포함할 수 있다. 예를 들어, 영구 스토리지(208)는 하드 디스크 드라이브, 솔리드-스테이트 드라이브, 플래시 메모리, 재기록 할 수 있는(rewritable) 광학 디스크, 재기록 할 수 있는 자기 테이프, 또는 이들의 일부 조합이 될 수 있다. 영구 스토리지(208)에서 사용하는 매체는 제거될 수 있다. 예를 들어, 제거가능(removable) 하드 드라이브는 영구 스토리지(208)에 사용할 수 있다.
- [0034] 키 오케스트레이션 에이전트(Key orchestration agent)(218)는 데이터 처리 시스템(200)을 위해, 데이터 암호화 키(data encryption key)(220), 씰링 키(sealing key)(222) 및 씰링 키(226)와 같은, 암호화 키들(cryptographic keys)을 생성하는 프로세스를 컨트롤한다. 데이터 암호화 키(220)는 애플리케이션(228)의 민감한 애플리케이션 데이터(232)를 암호화하고 해독하기 위한 암호화 키다. 씰링 키(222)는 키 오케스트레이션 에이전트(218)에 대해 고유하고(unique), 씰링 키(226)는 컨테이너(224)에 대해 고유하다. 각 씰링 키는, 예를 들어, 프로세서 유닛(204)과 같은, 데이터 처리 시스템(200)의 하드웨어에 기초할 수 있다. 씰링 키들(222 및 226)은 키 암호화 키들(key encryption keys)이다. 다시 말해, 키 오케스트레이션 에이전트(218)는 데이터 암호화 키(220)의 보안 및 보호를 위해 데이터 암호화 키(220)를 씰링하거나(seal) 암호화하도록(encrypt) 씰링 키(222)를 사용한다.

- [0035] 키 오케스트레이션 에이전트(218)는 데이터 처리 시스템(200) 내에 생성된 컨테이너(224)와 데이터 암호화 키(220)의 분산(distribution)을 조정하는 프로세스를 또한 컨트롤한다. 컨테이너(224)는, 포드들(pods)과 같은, 한 세트의 하나 또는 그 이상의 컨테이너들을 나타내며, 상기 컨테이너들은 애플리케이션(228)에 대응하는 한 세트의 하나 또는 그 이상의 워크로드들을 실행한다. 키 오케스트레이션 에이전트(218)는 데이터 암호화 키(220)를 컨테이너(224)와 공유하기 전에 로컬 증명 프로세스(a local attestation process)를 수행하여 컨테이너(224)가 인증되고(authenticated) 신뢰할 수 있는 컨테이너인지 확인한다. 데이터 암호화 키(220)를 수신한 이후, 컨테이너(224)에 대해 고유한 쉘링 키(226)로 컨테이너(224)는 데이터 암호화 키(220)를 쉘링하거나 암호화한다. 컨테이너(224)는 데이터 암호화 키(220)를 이용하여 애플리케이션(228)에 대응하는 민감한 애플리케이션 데이터(232)를 암호화 및 해독한다.
- [0036] 키 오케스트레이션 에이전트(218)는 한 세트의 하나 또는 그 이상의 다른 데이터 처리 시스템들(즉, 상기 클러스터 내의 한 세트의 다른 워커 노드들)과 데이터 암호화 키(220)의 분산을 조정하는 프로세스를 추가로 컨트롤한다. 키 오케스트레이션 에이전트(218)는 상기 클러스터의 각 워커 노드와 데이터 암호화 키(220)를 공유하기 전에 원격 증명 프로세스(a remote attestation process)를 수행하여 상기 세트의 다른 워커 노드들의 각 워커 노드가 인증되고 신뢰할 수 있는 워커 노드인지 확인한다.
- [0037] 애플리케이션(228)은 분산된, 다수-계층 애플리케이션(multi-tier application)을, 나타낼 수 있으며, 상기 다수-계층 애플리케이션은 데이터 처리 시스템(200) 상에서, 컨테이너(224)와 같은, 한 세트의 컨테이너들을 실행한다. 애플리케이션(228)은 모든 유형의 소프트웨어 애플리케이션을 나타낼 수 있고, 이들은, 예를 들어, 은행 애플리케이션, 금융 애플리케이션, 의료 애플리케이션, 보험 애플리케이션, 비즈니스 애플리케이션, 엔터테인먼트 애플리케이션, 게임 애플리케이션, 스트리밍 데이터 애플리케이션 등을 포함할 수 있으며, 클라이언트 디바이스들을 위한 서비스를 수행한다.
- [0038] 보안 구역(230)은 메모리(206)의 암호화된 섹션이며, 여기에서 애플리케이션(228)의 기능의 주요부분들이 보호되며, 민감한 애플리케이션 데이터(232)를 기밀로(confidential) 유지되고 수정되지 않은 상태로(unmodified) 유지된다. 민감한 애플리케이션 데이터(232)는 애플리케이션(228)에 대응하는 모든 유형의 개인정보를 나타내며, 이는, 예를 들어 개인 사용자 정보, 의료 기록들, 은행 잔고들, 신용 카드 번호들, 사회 보장 번호들, 재무 기록들 등과 같은 개인정보를 포함할 수 있다. 보안 구역(230)은 쉘링 키들(234)을 또한 저장한다. 쉘링 키들(234)은 쉘링 키(222)와 쉘링 키(226)를 나타낸다. 키 오케스트레이션 에이전트(218)와 컨테이너(224)는 데이터 암호화 키(220)를 쉘링하기 위해 보안 구역(230)로부터 필요한 때, 쉘링 키(222)와 쉘링 키(226)를, 각각, 검색한다(retrieve).
- [0039] 그 결과, 데이터 처리 시스템(200)은 특수 목적 컴퓨터 시스템으로 작동하는데, 여기서 데이터 처리 시스템(200)의 키 오케스트레이션 에이전트(218)는 데이터 처리 시스템(200) 내의 신뢰할 수 있는 컨테이너들과 그리고, 도 1의 클러스터(110)와 같은, 클러스터에서 다른 신뢰할 수 있는 워커 노드들과 함께 암호화 키 오케스트레이션을 가능하게 한다. 특히, 키 오케스트레이션 에이전트(218)는 데이터 처리 시스템(200)을 특수 목적 컴퓨터 시스템으로 변환하는데, 이는 키 오케스트레이션 에이전트(218)를 갖지 않는 현재 이용할 수 있는 범용 목적 컴퓨터 시스템들과 비교된다.
- [0040] 통신 유닛(210)은 도 1의 네트워크(102)와 같은, 네트워크를 통해 기타 컴퓨터들, 데이터 처리 시스템들 및 디바이스와의 통신을 제공한다. 통신 유닛(210)은 물리적 및 무선 통신 링크들 모두의 사용을 통해 통신들을 제공할 수 있다. 물리적 통신 링크는, 예를 들어, 유선, 케이블, 범용(universal) 직렬 버스, 또는 모든 기타 물리적 기술을 이용하여 데이터 처리 시스템(200)을 위한 물리적 통신 링크를 설정할 수 있다. 무선 통신 링크는, 예를 들어, 단파, 고주파, 초고주파, 마이크로파, 와이-파이(Wi-Fi), Bluetooth® 기술, 모바일 통신을 위한 글로벌 시스템(GSM), 코드 분할 다중 액세스(CDMA), 2세대(2G), 3세대(3G), 4세대(4G), 4G 롱 텀 에볼루션(LTE), LTE 어드밴스드, 5세대(5G), 또는 모든 기타 무선 통신 기술 또는 표준을 이용하여 데이터 처리 시스템(200)을 위해 무선 통신 링크를 설정할 수 있다.
- [0041] 입/출력 유닛(212)는 데이터 처리 시스템(200)에 연결될 수 있는 다른 디바이스와의 데이터의 입력과 출력을 허용한다. 예를 들어, 입/출력 유닛(212)은 키패드, 키보드, 마우스, 마이크, 및/또는 일부 기타 적절한 입력 디바이스를 통해 사용자 입력을 위한 연결을 제공할 수 있다. 디스플레이(214)는 사용자에게 정보를 표시하는 메커니즘을 제공하고, 예를 들어, 터치 스크린 기능을 포함하여 사용자가 사용자 인터페이스들 또는 입력 데이터를 통해 사용자가 온-스크린 선택들(on-screen selections) 수 있게 할 수 있다.
- [0042] 운영 시스템, 애플리케이션들, 및/또는 프로그램들에 대한 명령들은, 통신 패브릭(202)을 통해 프로세서

유닛(204)과 통신하는, 메모리(206) 및 영구 스토리지(208)에 위치할 수 있다. 이 예시적 예에서, 상기 명령들은 영구 스토리지(208) 상에서 기능적 형태로 있다. 이러한 명령들은 프로세서 유닛(204)에 의해 실행할 수 있도록 메모리(206)에 로드될 수 있다. 다른 실시 예들의 프로세스들은, 메모리(206)와 같은, 메모리에 위치할 수 있는, 컴퓨터-구현(computer-implemented) 명령들을 사용하여 프로세서 유닛(204)에 의해 수행할 수 있다. 이러한 프로그램 명령들은 프로그램 코드, 컴퓨터 사용 가능 프로그램 코드, 또는 컴퓨터 판독 가능 프로그램 코드로서 참조되며, 상기 프로그램 코드들은 프로세서 유닛(204)의 프로세서에 의해서 판독되고 실행될 수 있다. 다른 실시 예들에서, 상기 프로그램 명령들은 메모리(206) 또는 영구 스토리지(208)와 같은, 다른 물리적 컴퓨터 판독 가능 스토리지 디바이스들 상에서 구현될 수 있다.

[0043] 프로그램 코드(236)는, 프로세서 유닛(204)에 의한 실행을 위해 선택적으로 제거될 수 있고 데이터 처리 시스템(200)에 로드되거나 전송될 수 있는, 기능 형태로 컴퓨터 판독 가능 매체(238) 상에 위치해 있다. 프로그램 코드(236)와 컴퓨터 판독 가능 매체(238)는 컴퓨터 프로그램 제품(240)을 형성한다. 일 예에서, 컴퓨터 판독 가능 매체(238)는 컴퓨터 판독 가능 저장 매체(242) 또는 컴퓨터 판독 가능 신호 매체(244)가 될 수 있다. 컴퓨터 판독 가능 저장 매체(242)는, 예를 들어, 영구 스토리지(208)의 일부인, 하드 드라이브와 같은, 스토리지 디바이스로 전송하기 위해, 영구 스토리지(208)의 일부인, 드라이브 또는 기타 디바이스에 삽입 또는 배치되는 광 또는 자기 디스크를 포함할 수 있다. 컴퓨터 판독 가능 저장 매체(242)는 또한, 데이터 처리 시스템(200)에 연결되는, 하드 드라이브, 썸 드라이브, 또는 플래시 메모리와 같은, 영구 스토리지의 형태를 또한 취할 수도 있다. 일부 예들에서, 컴퓨터 판독 가능 저장 매체(242)는 데이터 처리 시스템(200)으로부터 제거 가능하지 않을 수 있다.

[0044] 이와 다르게, 프로그램 코드(236)는 컴퓨터 판독 가능 신호 매체를 사용하여 데이터 처리 시스템(200)으로 전송될 수 있다. 컴퓨터 판독 가능 신호 매체(244)는 예를 들어, 프로그램 코드(236)를 포함하는 전파된(propagated) 데이터 신호일 수 있다. 예를 들어, 컴퓨터 판독 가능 신호 매체(244)는 전자기(electromagnetic) 신호, 광학 신호, 및/또는 모든 기타 적절한 유형의 신호일 수 있다. 이러한 신호들은 무선(wireless) 통신 링크들, 광섬유 케이블, 동축 케이블, 유선(wire), 및/또는 모든 기타 적절한 유형의 통신 링크들과 같은, 통신 링크들을 통해서 전송될 수 있다. 다시 말해, 상기 통신 링크 및/또는 상기 연결은 예시적 예들에서 물리적 또는 무선일 수 있다. 컴퓨터 판독 가능 매체는 통신 링크들이나 상기 프로그램 코드가 포함된 무선 송신들과 같은, 무형의(non-tangible) 매체의 형태를 또한 취할 수 있다.

[0045] 일부 예시적 실시 예들에서, 프로그램 코드(236)는 데이터 처리 시스템(200) 내에서의 사용을 위해 컴퓨터 판독 가능 신호 매체(244)를 통해 다른 디바이스 또는 데이터 처리 시스템으로부터 네트워크를 통해서 영구 스토리지(208)에 다운로드 될 수 있다. 예를 들어, 데이터 처리 시스템의 컴퓨터 판독 가능 저장 매체에 저장된 프로그램 코드는 네트워크를 통해 상기 다른 데이터 처리 시스템으로부터 데이터 처리 시스템(200)으로 다운로드 될 수 있다. 프로그램 코드(236)를 제공하는 상기 데이터 처리 시스템은 서버 컴퓨터, 클라이언트 컴퓨터, 또는 프로그램 코드(236)를 저장하고 전송할 수 있는 일부 다른 디바이스일 수 있다.

[0046] 데이터 처리 시스템(200)을 위해 예시된 다른 컴포넌트들이 다른 실시 예들이 구현될 수 있는 방식에 아키텍처 제한을 제공하는 것을 의미하는 것은 아니다. 다른 예시적 실시 예들이 데이터 처리 시스템(200)을 위해 예시된 것들에 추가하여, 또는 대신하는 컴포넌트들을 포함한 데이터 처리 시스템에서도 구현될 수 있다. 도 2에 도시된 기타 컴포넌트들도 도시된 예시적 예들로부터 변경될 수 있다. 다른 실시 예들이 프로그램 코드를 실행할 수 있는 모든 하드웨어 디바이스 또는 시스템 케이블을 사용하여 구현될 수 있다. 일 예로서, 데이터 처리 시스템(200)은 무기(inorganic) 컴포넌트들에 집적된 유기(organic) 컴포넌트들을 포함하고/또는 인간을 제외한 유기 컴포넌트들로 완전히 구성될 수 있다. 예를 들어, 스토리지 디바이스는 유기 반도체로 구성될 수 있다.

[0047] 다른 예로서, 데이터 처리 시스템(200)에서 컴퓨터 판독 가능 스토리지 디바이스는 데이터를 저장할 수 있는 모든 하드웨어 장치(apparatus)이다. 메모리(206), 영구 스토리지(208) 및 컴퓨터 판독 가능 저장 매체(242)는 실제 형태의 물리적 스토리지 디바이스들의 예들이다.

[0048] 또 다른 예에서, 버스 시스템이 통신 패브릭(202)을 구현하기 위해 사용될 수 있고 시스템 버스 또는 입/출력 버스와 같은, 하나 또는 그 이상의 버스들로 구성될 수 있다. 물론, 버스 시스템은 모든 적절한 유형의 아키텍처를 사용하여 구현될 수 있으며, 상기 아키텍처는 버스 시스템에 부착된 다른 컴포넌트들이나 디바이스들 사이에서 데이터를 전송할 수 있다. 또한, 통신 유닛은 모듈이나 네트워크 어댑터와 같은, 데이터를 송신하고 수신하는 데 사용되는 하나 또는 그 이상의 디바이스들을 포함할 수 있다. 또한, 메모리는, 예를 들어, 메모리(206)이거나 인터페이스와 메모리 컨트롤러 허브와 같은 캐시가 될 수 있으며, 상기 인터페이스와 메모리 컨

트롤러 허브는 통신 패브릭(202)에 존재할 수 있다.

- [0049] 예시적 실시 예들은 오픈 소스 컨테이너 오케스트레이션 프레임워크들의 인프라스트럭처들에서 신뢰할 수 있는 컨테이너들과 워커 노드들 사이의 암호화 키 오케스트레이션 프로세스를 사용하여 동적, 분산 오픈 소스 컨테이너 오케스트레이션 프레임워크들에서, 민감한 애플리케이션 데이터를 공유하는 문제들을(challenges) 해결한다. 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 워커 노드들의 수가 "n"인 클러스터를 가질 때, 예시적 실시 예들은 클러스터 내의 모든 워커 노드가 데이터 암호화 키를 사용하여 상기 민감한 애플리케이션 데이터를 암호화 및 해독함에 의해서 보안 구역 내의 민감한 애플리케이션 데이터를 액세스하는 것을 가능하게 한다. 예시적 실시 예들은 신뢰할 수 있는(즉, 증명된) 워커 노드들 사이에서 데이터 암호화 키를 공유한다. 각 워커 노드는, 주어진 워커 노드의 보안 구역에 대해 고유한, 쉘링 키로 데이터 암호화 키를 암호화함에 의해 데이터 암호화 키를 보호한다. 이 메커니즘을 이용함에 의해서, 예시적 실시 예들은 하드웨어(즉, 워커 노드의 프로세서)에 대해 보안을 태그하고(tag), 또한 신뢰할 수 있는 워커 노드들 사이의 키 오케스트레이션의 메커니즘을 제공한다.
- [0050] 클러스터에 워커 노드를 생성하면, 상기 워커 노드는 상기 워커 노드의 보안 구역에서 실행되는 민감한 애플리케이션 데이터를 암호화하기 위한 데이터 암호화 키를 생성한다. 상기 워커 노드가 상기 클러스터의 유일한 워커 노드일 때, 상기 워커 노드에 위치한, 예를 들어, 데몬 세트(a daemon set)의 "키 오케스트레이션 데몬"과 같은, 키 오케스트레이션 소프트웨어 에이전트가, 상기 데이터 암호화 키를 생성한다. 데몬 세트는 특정 컨테이너의 인스턴스가 클러스터 내의 모든 워커 노드들, 또는 하나의 선택된 워커 노드 상에서 실행하는 것을 보장한다. 데몬 세트는 추가된 각 워커 노드에 컨테이너들을 생성하고 워커 노드들이 클러스터로부터 제거될 때 가비지(garbage)가 컨테이너들을 수집한다. 데몬 세트는 예를 들어, 클러스터 스토리지 데몬, 로그 수집기 데몬, 모니터링 데몬 등과 같은, 다른 데몬들을 포함할 수 있다.
- [0051] 워커 노드는 데이터 암호화 키를 사용하여 보안 구역에서 실행되는 민감한 애플리케이션 데이터를 보호한다. 또한, 워커 노드는 자신의 쉘링 키를 사용하여 데이터 암호화 키를 쉘링하며, 상기 쉘링 키는 워커 노드의 프로세서로부터 유래된 것이고, 상기 워커 노드에 대해 상기 쉘링 키가 고유하게 되게 한다. 보안 구역은 동일 워커 노드 상에서만 상기 쉘링 키를 검색할 수 있음에 주목해야 한다. 다시 말하면, 다른 워커 노드 상의 동일 보안 구역은 다른 쉘링 키를 갖게될 것이다.
- [0052] 새 워커 노드가 클러스터에 생성될 때, 새 워커 노드의 소프트웨어 에이전트(예: 키 오케스트레이션 데몬)는 클러스터의 기존 워커 노드로 원격 증명(a remote attestation)을 수행한다. 증명은 소프트웨어 실행 파일(executable)이 플랫폼 상에서 제대로 인스턴스화되었는지(properly instantiated) 입증하는 프로세스이다. 증명은 원격 당사자(a remote party)가 의도한 소프트웨어가 Intel® SGX 지원(enabled) 플랫폼 상의 보안 구역 내에서 안전하게 실행하고 있다는 확신을 얻게 해 준다. 증명은 주장으로(in an assertion) 다음 정보를 전달한다: 1) 증명되는 소프트웨어의 신원들(identities); 2) 측정되지 않은 상태(예: 소프트웨어가 실행 중인 모드)의 상세 정보; 및 3) 소프트웨어 그 자체와 연관되는 데이터. 증명에 성공하면, 새 워커 노드는 클러스터의 기존 워커 노드로부터 데이터 암호화 키의 사본을 수신한다. 데이터 암호화 키를 수신한 후, 새 워커 노드는 자신의 고유 쉘링 키로 데이터 암호화 키를 쉘링한다. 예시적 실시 예들은, 데이터 암호화 키와 암호화된 데이터를 포함하는, 공유 데이터베이스 또는 데이터 스토리지를 또한 사용할 수 있다. 신뢰할 수 있는(즉, 증명된) 워커 노드만이 공유 데이터베이스에 액세스할 수 있다.
- [0053] 따라서, 예시적 실시 예들은 클러스터 내의 다른 워커 노드들 상에서 분산 애플리케이션을 실행하는 다른 컨테이너들이 쉘링 키로 암호화 되어있는 데이터 암호화 키가 사용하여 다른 컨테이너들 사이에서 민감한 애플리케이션 데이터를 공유할 수 있게 한다. 각 컨테이너와 각 키 오케스트레이션 에이전트는 자신의 고유 쉘링 키를 갖는다. 각 키 오케스트레이션 에이전트가 증명 프로세스를 수행한다. 그 결과, 예시적 실시 예들은 컨테이너들이 워커 노드들의 클러스터 내에서 이동(예: 생성, 삭제 등)할 때에도, 민감한 애플리케이션 데이터는 신뢰할 수 있는(즉, 증명된) 컨테이너들 사이에서만 공유되도록 보장한다. 더욱이, 예시적 실시 예들은, 예를 들어, 하드 디스크, 파일 스토리지 등과 같은, 모든 형태의 스토리지에서 민감한 애플리케이션 데이터를 보호할 수 있다.
- [0054] 따라서, 예시적 실시 예들은 하나 또는 그 이상의 기술적 솔루션들을 제공하며, 상기 기술적 솔루션들은 오픈 소스 컨테이너 오케스트레이션 프레임워크 내의 컨테이너들 사이에서 민감한 애플리케이션 데이터를 공유함과 함께 기술적 문제를 극복한다. 그 결과, 이들 하나 또는 그 이상의 기술적 솔루션들은 데이터 보안의 분야에서 기술적 효과와 실질적인 애플리케이션을 제공한다.

- [0055] 이제 도 3을 참조하면, 암호화 키 오케스트레이션 시스템의 일 예를 예시하는 다이어그램이 일 예시적 실시 예에 따라 도시된다. 암호화 키 오케스트레이션 시스템(300)은 도 1의 네트워크 데이터 처리 시스템(100)과 같은, 데이터 처리 시스템들의 네트워크에서 구현될 수 있다. 암호화 키 오케스트레이션 시스템(300)은 오픈 소스 컨테이너 오케스트레이션 프레임워크의 멀티-노드(multi-node) 클러스터 내의 신뢰할 수 있는 컨테이너들과 노드들 사이에서 암호화 키 오케스트레이션을 제공하기 위한 하드웨어 및 소프트웨어 컴포넌트들의 시스템이다.
- [0056] 이 예에서, 암호화 키 오케스트레이션 시스템(300)은 오픈 소스 컨테이너 오케스트레이션 프레임워크(302), 클러스터(304) 및 데이터베이스(306)를 포함한다. 오픈 소스 컨테이너 오케스트레이션 프레임워크(302), 클러스터(304), 데이터베이스(306)는, 예를 들어, 도 1의 오픈 소스 컨테이너 오케스트레이션 프레임워크(112), 클러스터(110) 및 스토리지(108)일 수 있다. 그러나, 대안적인 예시적 실시 예에서, 데이터베이스(306)가 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)의 외부에 원격으로 위치할 수 있다는 점에 유의해야 한다. 또 다른 대안적인 예시적 실시 예에서, 데이터베이스(306)는 선택 사항(optional)이다.
- [0057] 이 예에서, 클러스터(304)는, 예를 들어, 도 1의 서버(104) 및 서버(106)와 같이, 워커 노드 A(308)와 워커 노드 B(310)를 포함한다. 그러나, 클러스터(304)는 여러 개의 워커 노드들을 포함할 수 있다. 워커 노드 A(308)와 워커 노드 B(310)는 도 1의 네트워크(102)와 같이, 네트워크(312)를 통해 서로 통신한다.
- [0058] 워커 노드 A(308)는 에이전트 A(314)와 컨테이너 1(316)을 포함한다. 마찬가지로, 워커 노드 B(310)는 에이전트 B(318)와 컨테이너 2(320)를 포함한다. 에이전트 A(314)와 에이전트 B(318)는 도 2의 키 오케스트레이션 에이전트(218)와 유사하다. 또한, 컨테이너 1(316)과 컨테이너 2(320)는 도 2의 컨테이너(224)와 유사하다.
- [0059] 에이전트 A(314)는 쉘링 키 A(SK-A)(322) 및 데이터 암호화 키(DEK)(324)를 포함한다. 쉘링 키 A(322)와 데이터 암호화 키(324)는, 예를 들어, 도 2에서 쉘링 키(222)와 데이터 암호화 키(220) 일 수 있다. 쉘링 키 A(322)는 에이전트A(314)에 대해서 고유하다. 에이전트(A314)는 쉘링 키 A(322)를 사용하여 데이터 암호화 키(324)를 쉘링한다.
- [0060] 컨테이너 1(316)은 데이터 암호화 키(DEK)(326)와 쉘링 키 1(SK-1)(328)을 포함한다. 쉘링 키 1(328)은 컨테이너 1(316)에 대해서 고유하다. 컨테이너 1(316)은 쉘링 키 1(328)을 사용하여 데이터 암호화 키(326)를 쉘링한다. 컨테이너 1(316)의 데이터 암호화 키(326)가 에이전트 A(314)의 데이터 암호화 키(324)와 동일하다는 것에 유의해야 한다. 에이전트 A(314)는 로컬 증명(330)에 기초하여 컨테이너 1(316)이 신뢰할 수 있는 컨테이너임을 결정한 후 컨테이너 1(316)과 함께 데이터 암호화 키(324)의 사본을 공유한다. 로컬 증명(330)은 워커 노드A(308)에서 실행되는 보안 구역들 사이에서 주장들(assertions)을 제공한다. 다시 말해, 워커 노드 A(308)상의 보안 구역들은 그들의 리포트들을 사용하여 서로에게 인증을 제공한다. 로컬 및 원격 증명 모두를 위해, 애플리케이션의 보호된 부분들이 보안 구역에 로드되며, 여기서 상기 애플리케이션의 코드와 데이터를 측정한다. 다음 리포트를 생성한다. 진정성 확정 후, 보안 구역들은 보안 채널을 통해 정보를 교환한다.
- [0061] 에이전트 B(318)는 쉘링 키 B(SK-B)(332)를 포함한다. 에이전트 A(314)의 데이터 암호화 키(324)가 에이전트 B(318)의 데이터 암호화 키(334)와 동일함에 주목해야 한다. 에이전트 A(314)는 에이전트 B(318)가 원격 증명(342)에 기초하여 신뢰할 수 있는 키 오케스트레이션 에이전트임을 결정한 후 에이전트 B(318)와 함께 데이터 암호화 키(324)의 사본을 공유한다. 원격 증명(342)은 에이전트 A(314)에서 에이전트 B(318)까지의 주장들을 제공한다. 원격 증명에서, 에이전트A(314)의 보안 구역은 에이전트 B(318)에게 인용(quote)을 보낸다. 그런 다음 에이전트 B(318)는 인증된(authentic) 프로세서가 상기 인용을 생성했는지에 관한 확인(validation)을 수행을 한다. 확인이 완료되면(Upon validation), 에이전트B(318)는 에이전트 A(314) 상의 보안 구역의 인증(authenticity)을 신뢰한다. 데이터 암호화 키(334)를 수신한 후, 에이전트 B(318)는 에이전트 B(318)에 대해서 고유한 쉘링 키 B(332)를 사용하여 데이터 암호화 키(334)를 쉘링한다.
- [0062] 컨테이너 2(320)는 데이터 암호화 키(DEK)(336) 및 쉘링 키 2(SK-2)(338)를 포함한다. 쉘링 키 2(338)는 컨테이너 2(320)에 대해서 고유하다. 컨테이너 2(320)는 쉘링 키 2(338)를 사용하여 데이터 암호화 키(336)를 쉘링한다. 컨테이너 2(320)의 데이터 암호화 키(336)가 에이전트 B(318)의 데이터 암호화 키(334)와 동일함에 유의해야 한다. 에이전트 B(318)는 컨테이너 2(320)가 로컬 증명(340)에 기초하여 신뢰할 수 있는 컨테이너임을 결정한 후 컨테이너 2(320)와 함께 데이터 암호화 키(334)의 사본을 공유한다. 로컬 증명(340)은 로컬 증명(330)과 유사한 로컬 증명 프로세스이다.
- [0063] 컨테이너 1(316)과 컨테이너 2(320)는 각각의 데이터 암호화 키들을 이용하여 컨테이너 1(316)과 컨테이너

너 2(320) 상에서 실행되는 분산 애플리케이션과 연관된 워크로드들에 대응하는 민감한 애플리케이션 데이터를 해독하고 암호화한다. 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)는 데이터베이스(306)를 이용하여, 예를 들어, 암호화된 데이터와 암호화 키들을 저장한다. 데이터베이스(306)는 공유 데이터베이스이다.

[0064] 이제 도 4를 참조하면, 워커 노드와 상기 워커 노드 상에 생성된 컨테이너 사이의 암호화 키 오케스트레이션을 위한 프로세스를 예시하는 플로차트가 일 예시적 실시 예에 따라 도시된다. 도 4에 도시된 프로세스는, 예를 들어, 도 1의 오픈 소스 컨테이너 오케스트레이션 프레임워크(112) 또는 도 3의 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 구현될 수 있다.

[0065] 상기 프로세스는 오픈 소스 컨테이너 오케스트레이션 프레임워크 내의 한 세트의 하나 또는 그 이상의 프로세서들이 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크 내의 키 오케스트레이션 에이전트를 갖는 워커 노드를 생성할 때 시작한다(단계 402). 상기 세트의 하나 또는 그 이상의 프로세서들은 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크의 하나 또는 그 이상의 노드들에 위치할 수 있다. 또한, 상기 세트의 프로세서들은, 상기 워커 노드의 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를 생성한다(단계 404). 또한, 상기 세트의 프로세서들은, 상기 키 오케스트레이션 에이전트를 사용하여, 상기 워커 노드의 보안 구역으로부터 상기 키 오케스트레이션 에이전트의 고유한 쉘링 키를 검색한다(retrieve)(단계 406). 상기 세트의 프로세서들은, 상기 키 오케스트레이션 에이전트의 고유한 쉘링 키를 사용하여, 데이터 암호화 키를 쉘링한다(단계 408).

[0066] 상기 세트의 프로세서들은 컨테이너가 상기 워커 노드에 생성되었는지를 결정한다(단계 410). 만약 상기 세트의 프로세서들이 컨테이너를 상기 워커 노드에 생성되지 않았다고 결정하면, 단계(410)의 출력은 아니오(no)이고, 상기 프로세스는 상기 세트의 프로세서들이 컨테이너가 상기 워커 노드에서 생성될 때까지 기다리는 단계(410)로 되돌아간다. 만약 상기 세트의 프로세서들이 컨테이너를 워커 노드에 생성되었다고 결정하면, 단계(410)의 출력은 예(yes)이고, 상기 세트의 프로세서들은, 상기 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너의 진정성(authenticity)을 확정하기 위해 상기 컨테이너로 로컬 증명(local attestation)을 수행한다(단계 412).

[0067] 상기 세트의 프로세서들은 상기 로컬 증명에 기초하여 상기 컨테이너가 신뢰할 수 있는지를 결정한다(단계 414). 만약 상기 세트의 프로세서들이 상기 로컬 증명에 기초하여 상기 컨테이너를 신뢰할 수 없다고 결정하면, 단계(414)의 출력은 아니오(no)이고, 상기 세트의 프로세서들은 상기 컨테이너에 의한 상기 데이터 암호화 키에 대한 액세스를 거부한다(단계 416). 그 후에, 상기 프로세스는 종료된다.

[0068] 만약 상기 세트의 프로세서들이 상기 로컬 증명에 기초하여 상기 컨테이너를 신뢰할 수 있다고 결정하면, 단계(414)의 출력은 예(yes)이고, 상기 세트의 프로세서들은, 상기 키 오케스트레이션 에이전트를 사용하여, 상기 데이터 암호화 키를 상기 컨테이너와 공유한다(단계 418). 또한, 상기 세트의 프로세서들은, 상기 컨테이너와 공유된 데이터 암호화 키를 사용하여, 상기 컨테이너의 워크로드에 대응하는 데이터를 암호화한다(단계 420). 더욱이, 상기 세트의 프로세서들은 상기 워커 노드의 식별자와 공유 데이터베이스에서 상기 컨테이너의 고유한 쉘링 키로 쉘링된 데이터 암호화 키를 저장한다(단계 422). 그 후에, 상기 프로세스는 종료된다.

[0069] 이제 도 5a-5b를 참조하면, 클러스터 내에서 기존 워커 노드와 새로 생성된 워커 노드 사이의 암호화 키 오케스트레이션을 위한 프로세스를 예시하는 플로차트가 일 예시적 실시 예에 따라 도시된다. 도 5a-5b에 도시된 프로세스는, 예를 들어, 도 1의 오픈 소스 컨테이너 오케스트레이션 프레임워크(112) 또는 도 3의 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 구현할 수 있다.

[0070] 상기 프로세스는, 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 한 세트의 하나 또는 그 이상의 프로세서들이 제1 키 오케스트레이션 에이전트를 갖는 제1 워커 노드가 상기 세트의 프로세서들에 의해 상기 클러스터 내에서 생성된 후 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크의 노드 클러스터 내에 제2 키 오케스트레이션 에이전트를 갖는 제2 워커 노드를 생성할 때, 시작한다(단계 502). 상기 세트의 프로세서들은, 상기 제2 워커 노드의 상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제2 키 오케스트레이션 에이전트의 진정성을 확정하기 위해 상기 제1 워커 노드의 상기 제1 키 오케스트레이션 에이전트로 원격 증명을 수행한다(단계(504)).

[0071] 상기 세트의 프로세서들은 상기 제2 키 오케스트레이션 에이전트가 상기 원격 증명에 기초하여 신뢰할 수 있는지를 결정한다(단계 506). 만약 상기 세트의 프로세서들이 상기 제2 키 오케스트레이션 에이전트를 상기 원

격 증명에 기초하여 신뢰할 수 없다고 결정하면, 단계(506)의 출력은 아니오(no)이고, 상기 세트의 프로세서들은 상기 제2 키 오के스트레이션 에이전트에 의한 상기 제1 키 오के스트레이션 에이전트의 데이터 암호화 키에 대한 액세스를 거부한다(단계 508). 그 후에, 상기 프로세스는 종료된다.

[0072] 만약 상기 세트의 프로세서들이 상기 제2 키 오के스트레이션 에이전트를 상기 원격 증명에 기초하여 신뢰할 수 있다고 결정하면, 단계(506)의 출력은 예(yes)이고, 상기 세트의 프로세서들은, 상기 제1 워커 노드의 상기 제1 키 오के스트레이션 에이전트를 사용하여, 상기 제2 키 오के스트레이션 에이전트와 상기 제1 키 오के스트레이션 에이전트의 데이터 암호화 키를 공유한다(단계 510). 또한, 상기 세트의 프로세서들은, 상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 제2 워커 노드의 보안 구역으로부터 상기 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를 검색한다(단계 512). 그 다음, 상기 세트의 프로세서들은, 상기 제2 키 오케스트레이션 에이전트의 고유한 쉘링 키를 사용하여, 상기 데이터 암호화 키를 쉘링한다(단계 514).

[0073] 상기 세트의 프로세서들은 컨테이너가 상기 제2 워커 노드에 생성되었는지를 결정한다(단계 516). 만약 상기 세트의 프로세서들이 컨테이너가 상기 제2 워커 노드에 생성되지 않았다고 결정하면, 단계(516)의 출력은 아니오(no)이고, 상기 프로세스는 상기 세트의 프로세서들이 컨테이너가 상기 제2 워커 노드에 생성되도록 기다리는 단계(516)로 되돌아간다. 만약 상기 세트의 프로세서들이 컨테이너가 상기 제2 워커 노드에 생성되었다고 결정하면, 단계(516)의 출력은 예(yes)이고, 상기 세트의 프로세서들은, 상기 제2 키 오케스트레이션 에이전트를 사용하여, 상기 컨테이너의 진정성을 확정하기 위해 상기 제2 워커 노드의 컨테이너로 로컬 증명을 수행한다(단계 518).

[0074] 상기 세트의 프로세서들은 상기 제2 워커 노드의 컨테이너가 상기 로컬 증명에 기초하여 신뢰할 수 있는지를 결정한다(단계 520). 만약 상기 세트의 프로세서들이 상기 제2 워커 노드의 컨테이너를 상기 로컬 증명에 기초하여 신뢰할 수 없다고 결정하면, 단계(520)의 출력은 아니오(no)이고, 상기 세트의 프로세서들은 상기 제2 워커 노드의 컨테이너에 의한 상기 제2 키 오케스트레이션 에이전트의 상기 데이터 암호화 키에 대한 액세스를 거부한다(단계 522). 그 후에, 상기 프로세스는 종료된다.

[0075] 만약 상기 세트의 프로세서들이 상기 제2 워커 노드의 컨테이너를 상기 로컬 증명에 기초하여 신뢰할 수 있다고 결정하면, 단계(520)의 출력은 예(yes)이고, 상기 세트의 프로세서들은, 상기 제2 키 오케스트레이션 에이전트를 사용하여, 데이터 암호화 키를 상기 제2 워커 노드의 컨테이너와 공유한다(단계 524). 또한, 상기 세트의 프로세서들은, 상기 제2 워커 노드의 컨테이너와 공유된 데이터 암호화 키를 사용하여, 상기 제1 워커 노드 상의 다른 컨테이너의 워크로드에 대응하는 데이터를 암호 해독하고, 상기 데이터에 관하여 한 세트의 하나 또는 그 이상의 연산들을 수행한다(단계 526). 상기 세트의 연산들은, 예를 들어, 생성, 읽기, 갱신 및 삭제 연산들 중 하나 또는 그 이상을 포함할 수 있다. 그 다음에, 상기 세트의 프로세서들은, 상기 제2 워커 노드의 컨테이너와 공유된 데이터 암호화 키를 사용하여, 상기 세트의 연산들을 수행한 후 상기 데이터를 암호화한다(단계 528). 그 다음, 상기 세트의 프로세서들은 상기 제2 워커 노드의 식별자와 공유 데이터베이스에서 상기 제2 워커 노드 상의 컨테이너의 고유한 쉘링 키로 쉘링된 데이터 암호화 키를 저장한다(단계 530). 그 후에, 상기 프로세스는 종료된다.

[0076] 이제 도 6을 참조하면, 워커 노드들의 데이터 암호화 키들을 보호하기 위한 프로세스를 예시하는 플로차트가 일 예시적 실시 예에 따라 도시된다. 도 6에 도시된 프로세스는, 예를 들어, 도 1의 오픈 소스 컨테이너 오케스트레이션 프레임워크(112) 또는 도 3의 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 구현할 수 있다.

[0077] 상기 프로세스는, 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크의 복수의 워커 노드들의 각 워커 노드가 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내에 있는 보안 구역에 위치한 공유된 민감한 애플리케이션 데이터에 액세스할 때, 시작한다(단계 602). 또한, 상기 복수의 워커 노드들의 각 워커 노드는 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 쉘링 키를 사용하여 각 워커 노드의 데이터 암호화 키를 보호한다(단계 604). 그 후에, 상기 프로세스는 종료된다.

[0078] 이제 도 7을 참조하면, 클러스터 내의 워커 노드들 사이에서 데이터 암호화 키를 공유하는 프로세스를 예시하는 플로차트는 일 예시적 실시 예에 따라 도시된다. 도 7에 도시된 상기 프로세스는, 예를 들어, 도 1의 오픈 소스 컨테이너 오케스트레이션 프레임워크(112) 또는 도 3의 오픈 소스 컨테이너 오케스트레이션 프레임워크(302)와 같은, 오픈 소스 컨테이너 오케스트레이션 프레임워크에서 구현될 수 있다.

[0079] 상기 프로세스는, 상기 오픈 소스 컨테이너 오케스트레이션 프레임워크가 상기 오픈 소스 컨테이너 오케

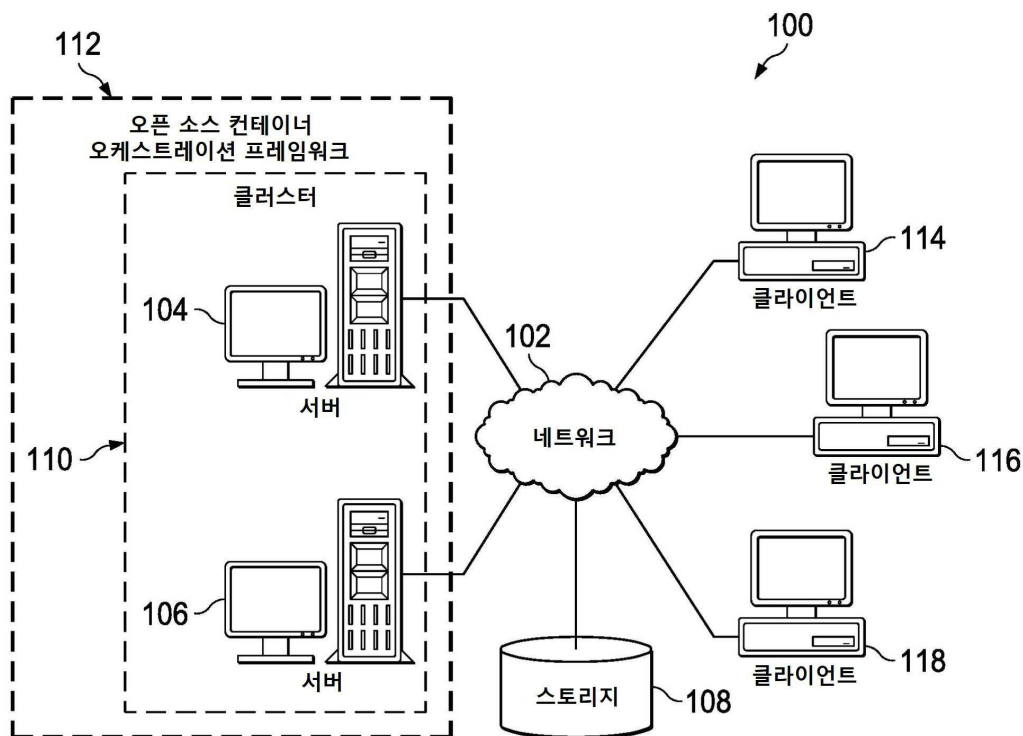
스트레이션 프레임워크의 클러스터에 제1 워커 노드를 설정할 때, 시작한다(단계 702). 상기 제1 워커 노드는 데이터 암호화 키를 생성한다(단계 704). 또한, 상기 제1 워커 노드는 상기 제1 워커 노드의 하드웨어에 기초하여 상기 제1 워커 노드 상의 보안 구역에 대해서 고유한 제1 쉘링 키를 생성하고, 상기 제1 쉘링 키를 사용하여 상기 데이터 암호화 키를 암호화함에 의해서 상기 데이터 암호화 키를 보호한다(단계 706).

[0080] 그 다음에, 상기 제1 워커 노드는 상기 클러스터에서 상기 제2 워커 노드의 생성에 응답하여 상기 클러스터의 제2 워커 노드로 원격 증명을 수행한다(단계 708). 상기 제1 워커 노드는 상기 원격 증명의 성공적 수행에 따라 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을 제공한다(단계 710). 상기 제2 워커 노드는 상기 제2 워커 노드의 하드웨어에 기초하여 상기 제2 워커 노드 상의 보안 구역에 대해서 고유한 제2 쉘링 키를 생성하고, 상기 제2 쉘링 키를 사용하여 상기 데이터 암호화 키를 암호화함에 의해서 상기 데이터 암호화 키를 보호한다(단계 712). 그 후에, 상기 프로세스는 종료된다.

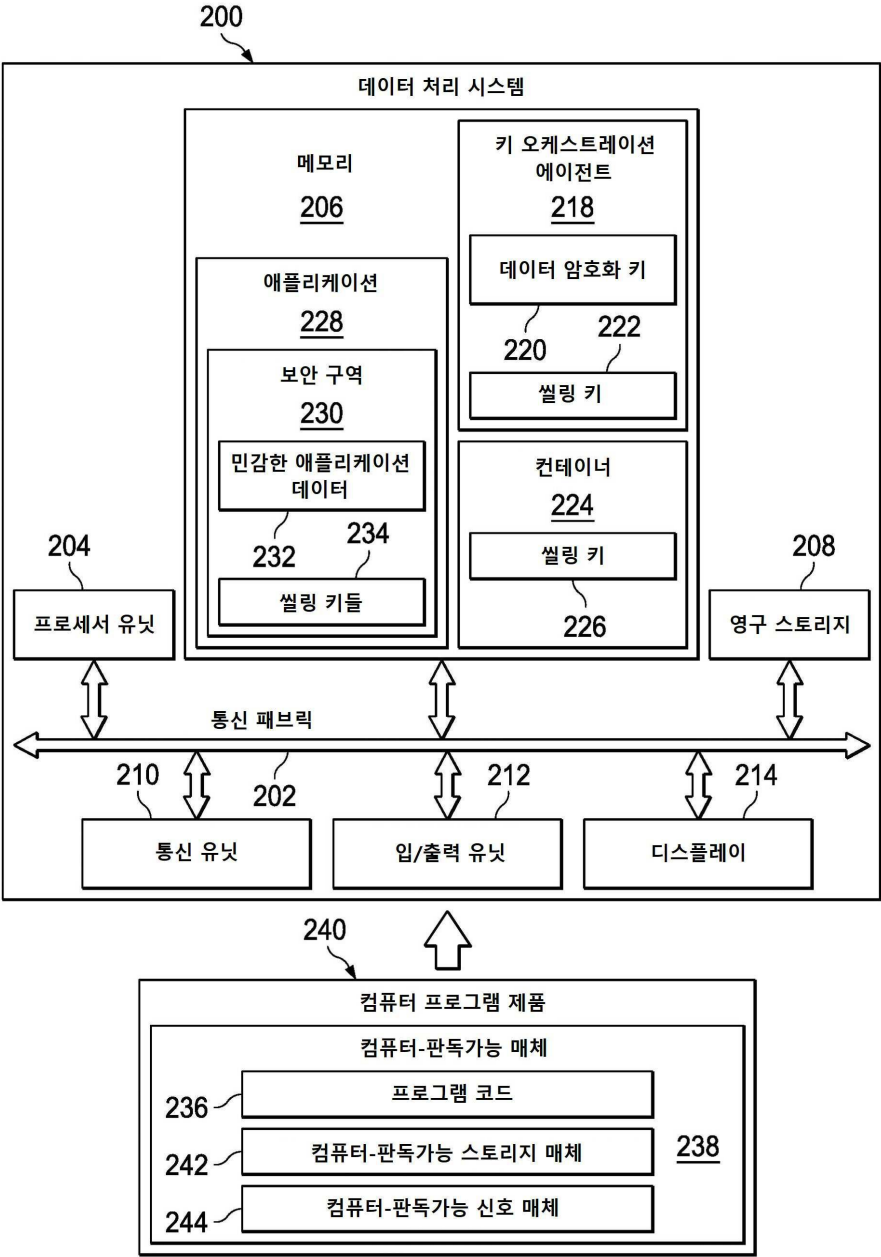
[0081] 따라서, 본 발명의 예시적 실시 예들은 오픈 소스 컨테이너 오케스트레이션 프레임워크의 멀티-노드 클러스터 내 신뢰할 수 있는 노드들과 컨테이너들 사이에서 암호화 키 오케스트레이션을 제공하기 위한 컴퓨터-구현 방법, 컴퓨터 시스템 및 컴퓨터 프로그램 제품을 제공한다. 본 발명의 다양한 실시 예들의 설명들은 예시의 목적으로 제공되는 것이지만, 개시되는 실시 예들이 전부라거나 한정하려는 의도가 있는 것은 아니다. 기술된 실시 예들의 범위와 정신으로부터 벗어남 없이, 많은 수정과 변형이 당업자들에게 명백할 것이다. 여기에 사용된 용어는 실시 예들의 원리들, 시장에서 발견된 기술들에 대한 실제 애플리케이션 또는 기술적 개선을 가장 잘 설명하거나, 또는 당업자들이 여기에 공개된 실시 예들을 이해할 수 있도록 하기 위해 선택되었다.

## 도면

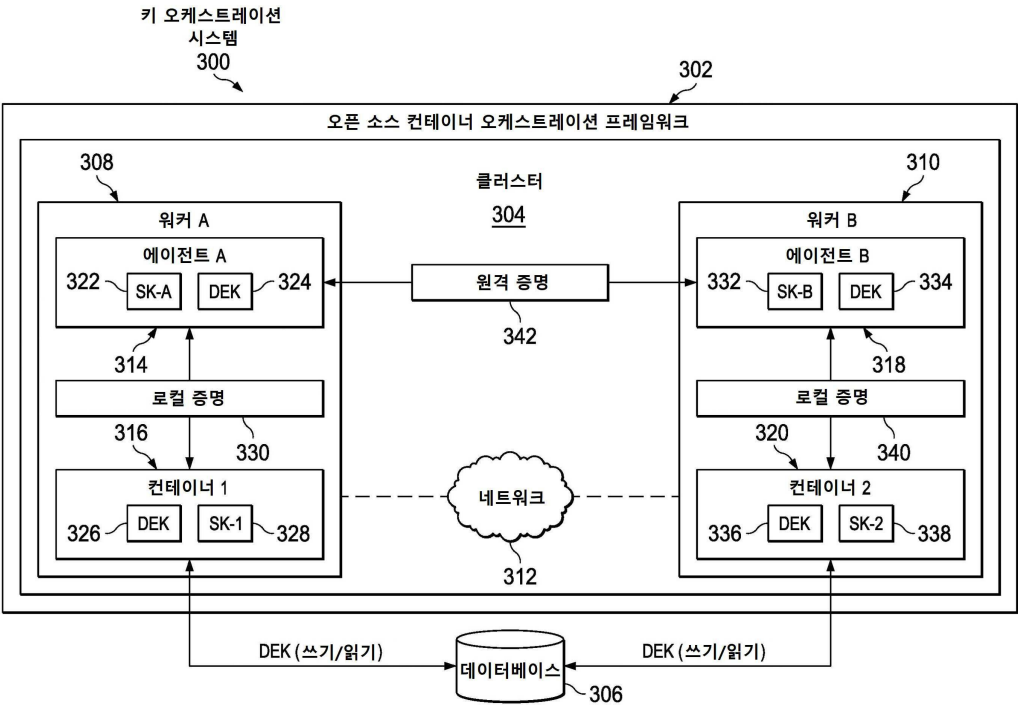
### 도면1



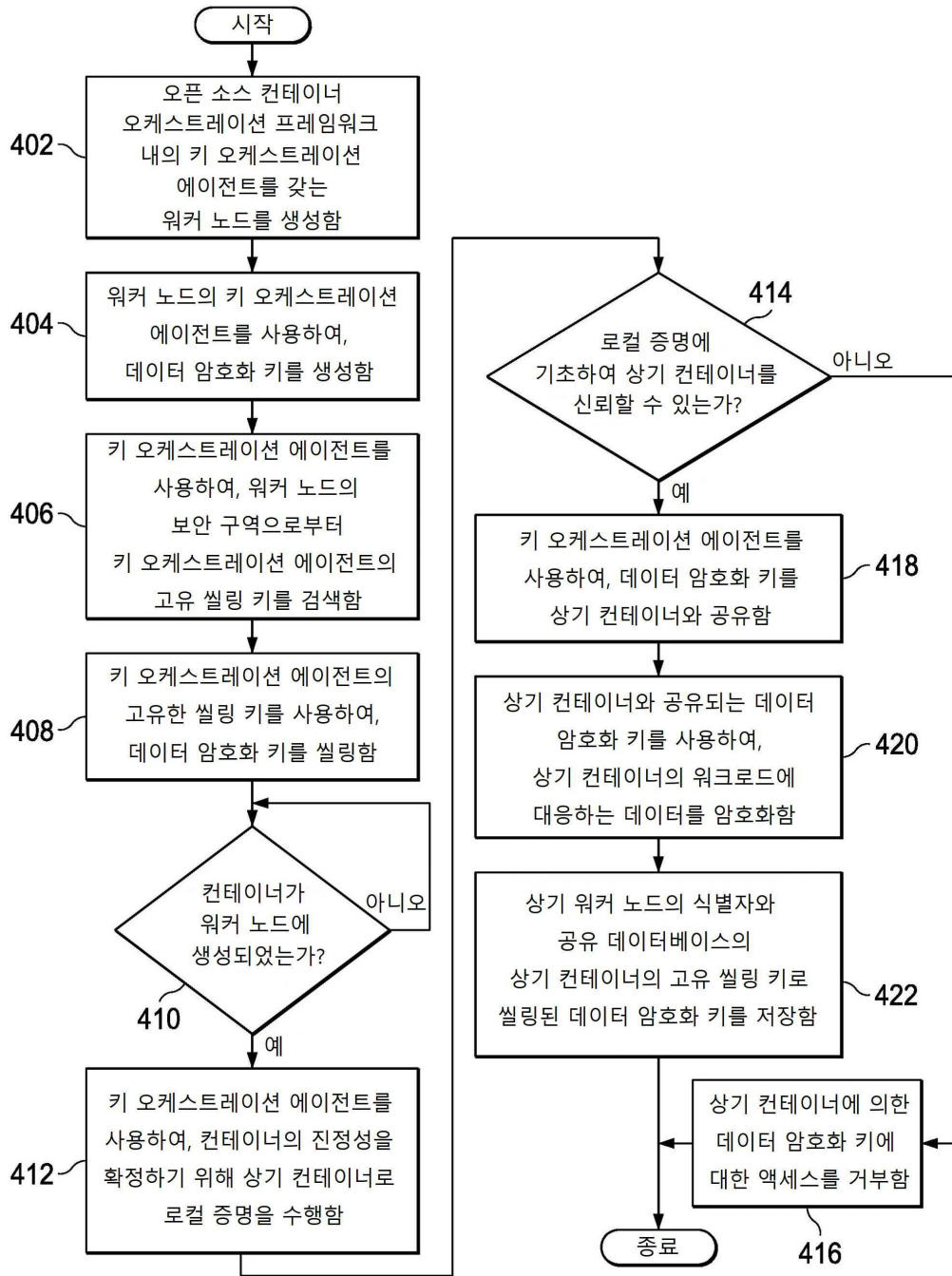
도면2



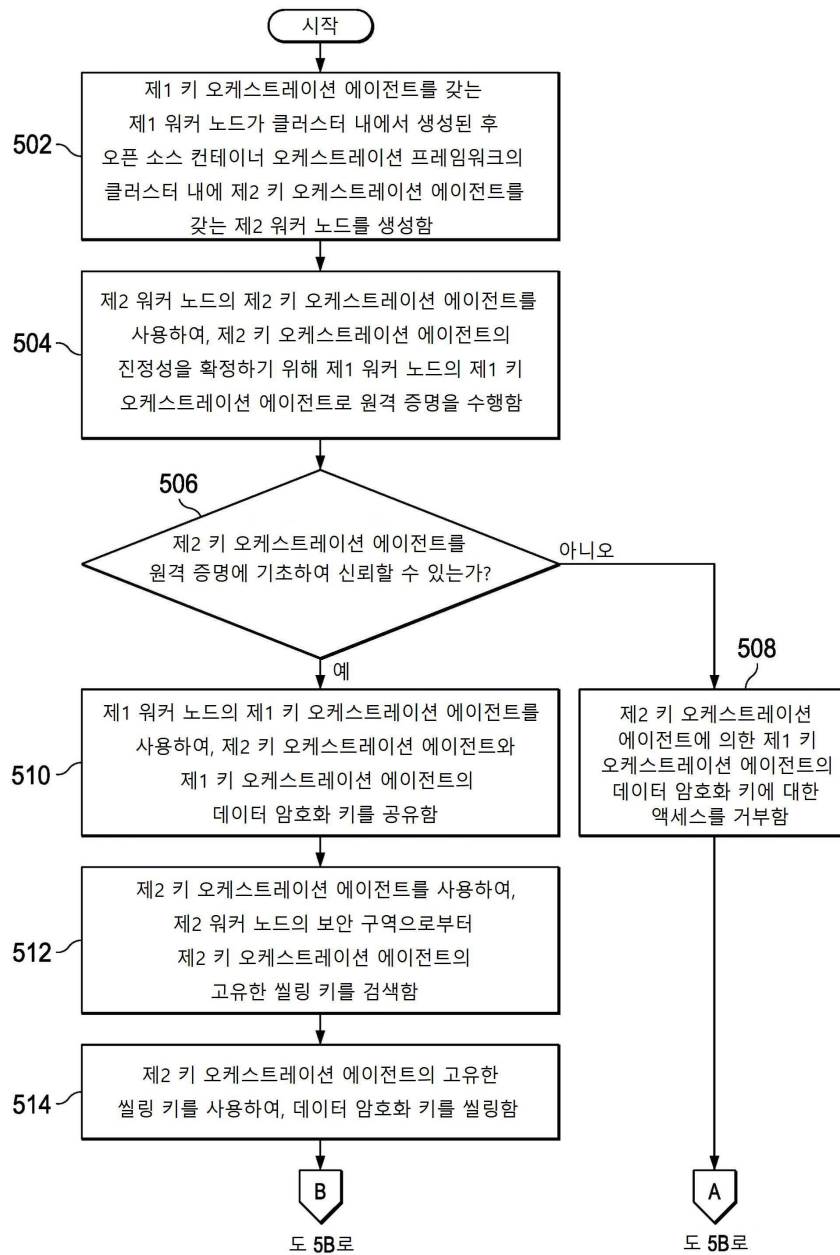
도면3



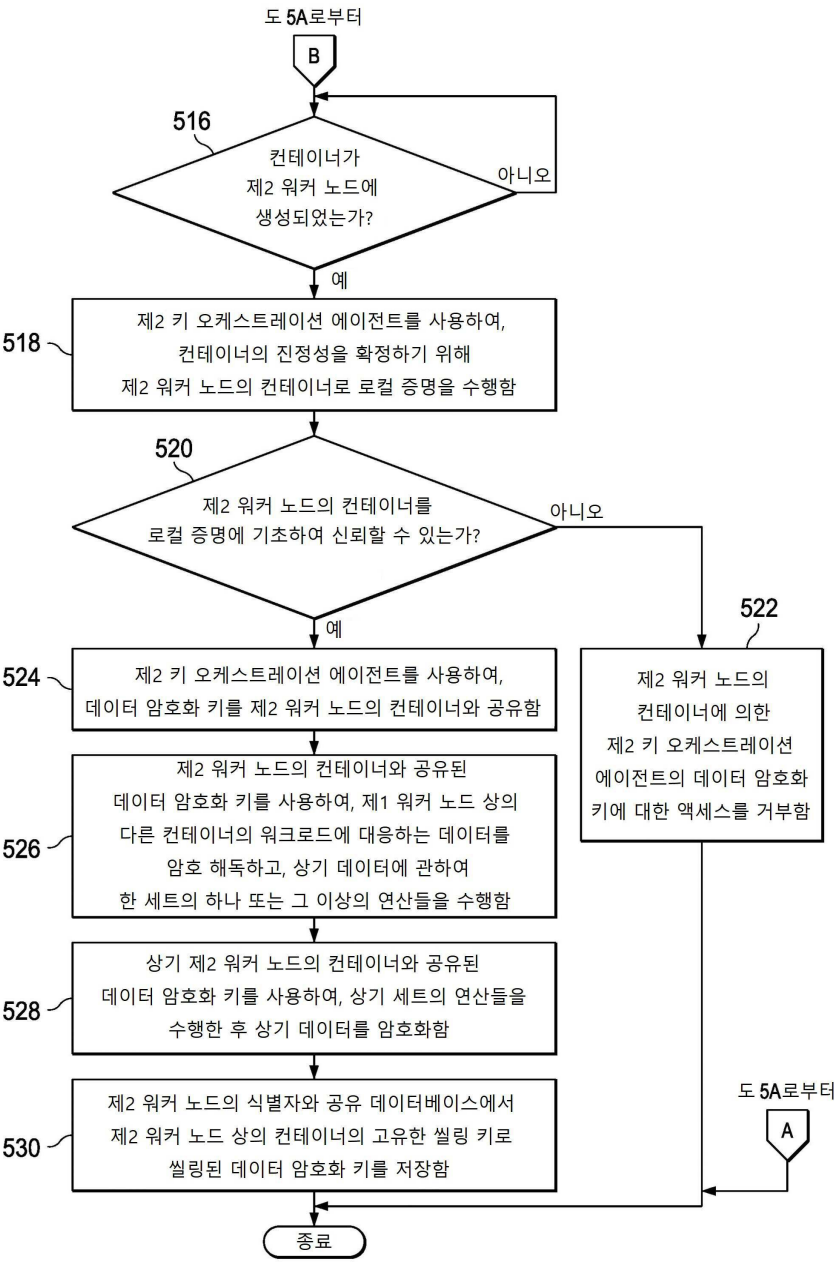
도면4



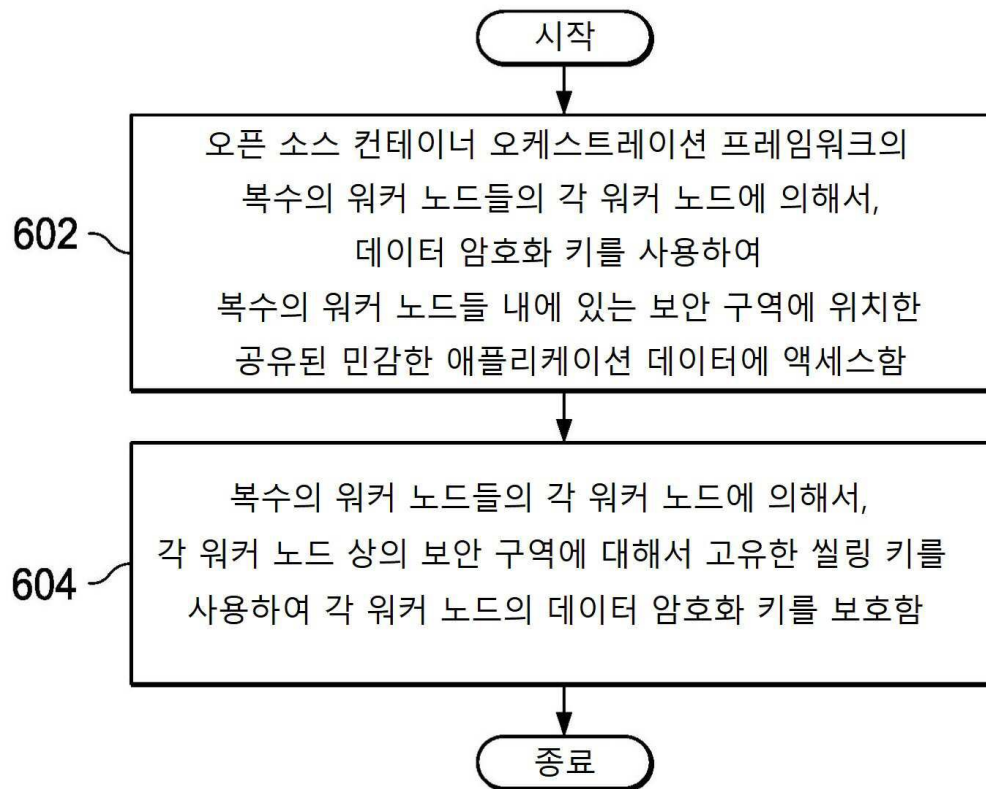
도면5a



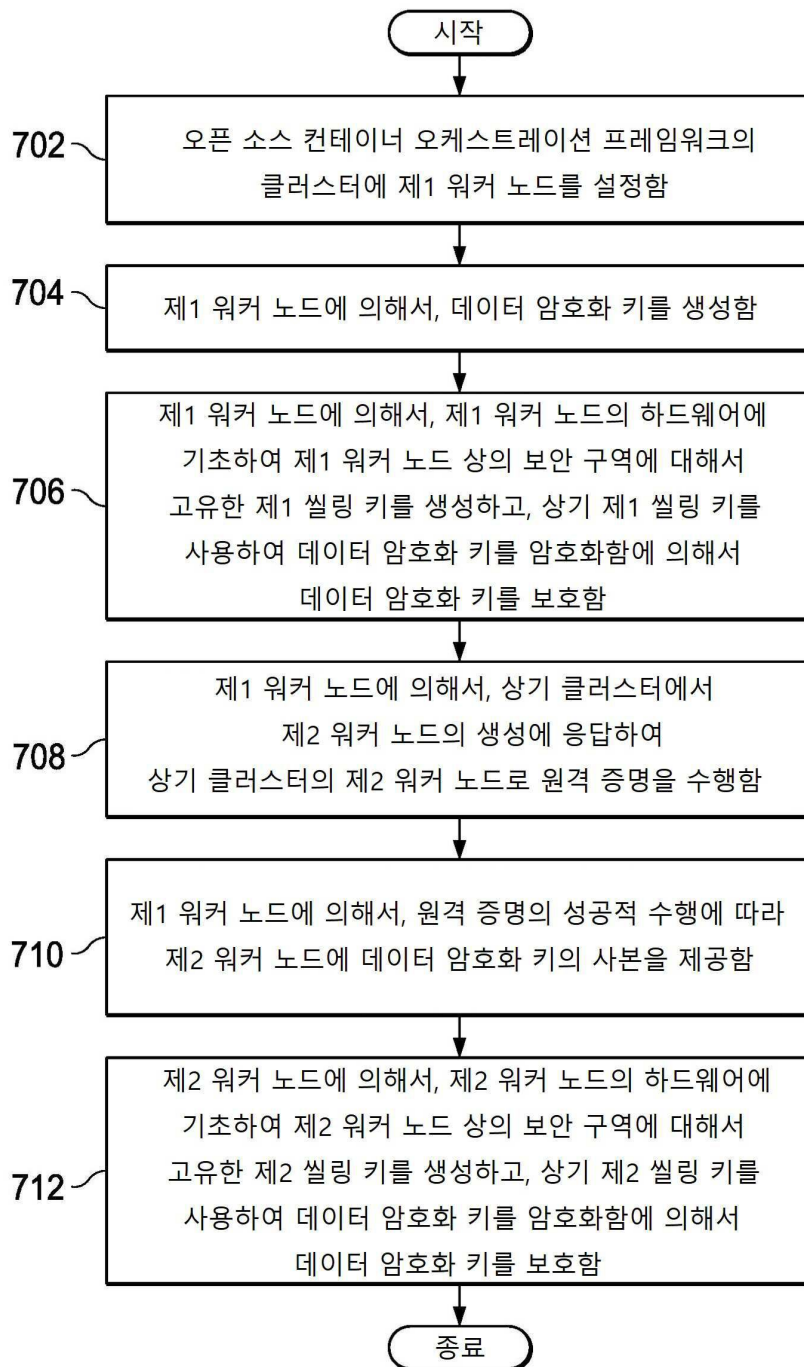
도면5b



도면6



도면7



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 2

【변경전】

관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법에 있어서, 상기 방법은:

복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared

sensitive application data)를, 액세스하는 단계;

상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계;

클러스터(a cluster)에 제1 워커 노드를 설정하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를, 생성하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를 보호하기 위해 상기 제1 워커 노드 상의 상기 보안 구역에 고유한 제1 씰링 키를, 생성하는 단계;

상기 클러스터에서 상기 제2 워커 노드의 생성에 응답하여, 상기 제1 워커 노드에 의해서, 상기 클러스터의 제2 워커 노드로 원격 증명(remote attestation)을, 수행하는 단계; 및

상기 원격 증명의 성공적 수행에 따라, 상기 제1 워커 노드에 의해서, 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을, 제공하는 단계를 포함하는

방법.

#### 【변경후】

관리되는 컨테이너 환경에서 공유된 민감한 정보를 액세스하기 위한 방법에 있어서, 상기 방법은:

복수의 워커 노드들(a plurality of worker nodes)의 각 워커 노드에 의해서, 데이터 암호화 키를 사용하여 상기 복수의 워커 노드들 내의 보안 구역(a secure enclave)에 위치한 공유된 민감한 애플리케이션 데이터(shared sensitive application data)를, 액세스하는 단계;

상기 복수의 워커 노드들의 각 워커 노드에 의해서, 각 워커 노드 상의 상기 보안 구역에 대해서 고유한 씰링 키(a sealing key)를 사용하여 각 워커 노드의 상기 데이터 암호화 키를, 보호하는 단계;

클러스터(a cluster)에 제1 워커 노드를 설정하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를, 생성하는 단계;

상기 제1 워커 노드에 의해서, 상기 데이터 암호화 키를 보호하기 위해 상기 제1 워커 노드 상의 상기 보안 구역에 고유한 제1 씰링 키를, 생성하는 단계;

상기 클러스터에서 제2 워커 노드의 생성에 응답하여, 상기 제1 워커 노드에 의해서, 상기 클러스터의 제2 워커 노드로 원격 증명(remote attestation)을, 수행하는 단계; 및

상기 원격 증명의 성공적 수행에 따라, 상기 제1 워커 노드에 의해서, 상기 제2 워커 노드에 상기 데이터 암호화 키의 사본을, 제공하는 단계를 포함하는

방법.